JEDE AUSGABE MIT FIRMENREGISTER UND PRAKTIKUMSBÖRSE

01/2011 € 6,90

making games MAGAZIN

MAGAZIN FÜR SPIELE-ENTWICKLUNG UND BUSINESS-DEVELOPMENT

TECHNOLOGIEN FÜR 2011

S3D, KINECT, CROSS-PLATFORM: WIE CRYTEK, RARE, BIGPOINT UND UBISOFT MONTREAL SCHON HEUTE DIE TECHNIK VON MORGEN ANWENDEN

GAME DESIGN
WIE MAN DAS BALANCING BEREITS IN DER KONZEPTPHASE ERLEDIGT

EXKLUSIV-INTERVIEW
DIE PROJEKTE UND PERSPEKTIVEN DES DEUTSCHEN ZYNGA-STUDIOS

PUBLISHING-DEALS
KONKRETE ZAHLEN: WAS MAN MIT FLASH GAMES WIRKLICH VERDIENT

DAS AUFLAGENSTÄRKSTE VERZEICHNIS DER DEUTSCHEN GAMES-BRANCHE

key players

MIT FREELANCER-RUBRIK!

Alle Infos: www.makinggames.de/keyplayers

BIS ZUM 31. MÄRZ 2011 BUCHEN UND SPAREN!

2-SEITIGES FIRMENPROFIL: 990 EURO (STATT 1.290 EURO)

FREELANCER-EINTRAG: 195 EURO (STATT 245 EURO)

ANNAHMESCHLUSS: 31. MAI 2011

Start Making Games
Gute Vorsätze? Jawoll!

Das Knifflige an den guten Vorsätzen für's neue Jahr ist, dass wir sie uns in der Regel selbst ausdenken, häufig nicht weitererzählen und damit auch die einzige Kontrollinstanz sind. Dabei wären wir doch viel motivierter/getriebener, wenn die guten Vorsätze vom Partner oder dem besten Freund kommen würden. Oder vom Leser. Vor drei Ausgaben haben wir Sie deshalb gefragt, was denn ein paar gescheite Vorsätze für unser Making Games Magazin wären, sprich was wir noch verbessern können. Für die rege Teilnahme an unserer großen Leserumfrage möchten wir uns an dieser Stelle noch einmal ganz herzlich bedanken. Hier ein paar der interessantesten Ergebnisse:

1. Insgesamt machen wir unseren Job glücklicherweise ganz ordentlich, die Durchschnittsnote für unser Magazin liegt bei 1,8. Am besten werden Verständlichkeit (1,63) und Kompetenz (1,73) beurteilt, noch Verbesserungsbedarf gibt's bei der Themenauswahl (2,13).
2. Jedes einzelne Heft wird im Schnitt von 1,7 Personen gelesen. Falls Sie es übrigens leid sind, sich immer ein Heft vom Kollegen ausleihen zu müssen: Machen Sie Ihren Chef doch mal auf unser unverschämt günstiges Firmenabo-Angebot aufmerksam! (Seite 70)
3. Gut, das war uns schon vorher klar: Unsere Leser sind ganz schön schlau – über 70% haben einen Universitätsabschluss.
4. Die größte Lesergruppe stellen die Programmierer (über 40%), danach folgen Game Designer (10%) Producer (7%) und Grafiker (5%).
5. In Folge von 4. wenig verwunderlich: Ganz oben auf dem Wunschzettel unserer Leser für das Making Games Magazin stehen Programmier- und Entwicklungs-Artikel.

Wir fangen dann schon mal an

Sylvester hin oder her: Mit der Making Games 01/11, die sie gerade in den Händen halten, hat für uns das neue Jahr schon jetzt begonnen. Entsprechend haben wir uns bemüht, unsere guten Vorsätze für 2011 bereits in dieser Ausgabe umzusetzen. Mit »Parallelisierung für die Xbox 360« (Seite 36), »Cross-Platform-Entwicklung« (Seite 26) und »Stereoskopie für Spiele-Entwickler« (Seite 12) gibt es gleich drei Artikel, die vor allem für Programmierer relevante Themen beleuchten. Außerdem möchten wir noch mehr in die Tiefe gehen als bisher – schließlich ist Making Games ein Heft von Profis für Profis. Ein schönes Beispiel für diesen Anspruch ist unser neuer Workshop »Testdokumentation« (Seite 40). Und zu guter Letzt wollen wir auch ein wenig kritischer werden und auch mal über die Schattenseiten der Spiele-Entwicklung berichten, etwa mit unserer neuen Rant-Serie »Wenn ihr wüsstet ...« (Seite 77).

Also zusammengefasst alles noch größer, besser und informativer – in etwa so wie bei unserem Making Games Wandkalender, der kostenlos dieser Ausgabe beiliegt und den wir gegenüber der Vorjahresversion noch einmal kräftig aufgemöbelt haben.

Wir wünschen Ihnen ein spannendes, kreatives und erfolgreiches 2011!

das Team von Making Games

Heiko Klinge

ist Chefredakteur vom Making Games Magazin.

WWW
makinggames.de

- Immer wenn Sie diesen Hinweis bei einem Artikel sehen, möchten wir Sie auf ergänzende oder vertiefende Informationsangebote unseres Online-Portals aufmerksam machen.

» Wir haben uns bemüht, unsere guten Vorsätze für 2011 bereits in dieser Ausgabe umzusetzen. «

01/2011
Inhalt

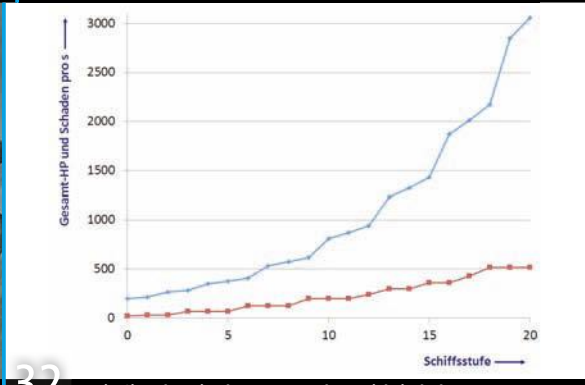
26 Wie bei Bigpoint die Cross-Platform-Entwicklung funktioniert.

58 Tau Petersson im Interview.

80 Ein Tag beim Sims Studio im sonnigen Kalifornien.

78 Was macht Richard Bartle?

12 Wie funktioniert die Stereoskopie in der CryEngine 3?



32 Wie Sie schon in der Konzeptphase richtig balancen.

Branche

06 **Branche / Köpfe**
Bemerkenswertes aus der Branche

07 **Eventkalender**
Wichtige Branchenveranstaltungen

08 **Passiert auf makinggames.de**
Interessantes aus dem World Wide Web

10 **Making Games Tools: Unity3D 3.0**
Die Meinung der Profis zu wichtigen Updates

Titelstory Technologien für 2011

12 **Best Practice**
Stereoskopie für Spiele-Entwickler
von Alexander Taube

20 **Interview**
Kinect aus Entwicklersicht
von Heiko Klinge

26 **Case Study**
Cross-Platform-Entwicklung
von Olliver Heins

Rubriken

03 **Editorial**

77 **Wenn ihr wüsstet ...**
warum Lokalisierung wirklich schiefgeht!

78 **Was macht eigentlich ...** Richard Bartle

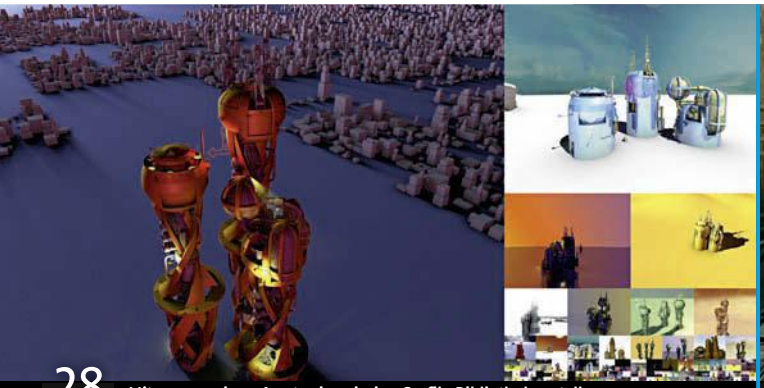
80 **Ein Tag beim ...** Sims Studio

82 **Vorschau**

82 **Impressum**

Grafik

28 **Best Practice**
Automatisierte Content-Generierung
von Jochen Winkler

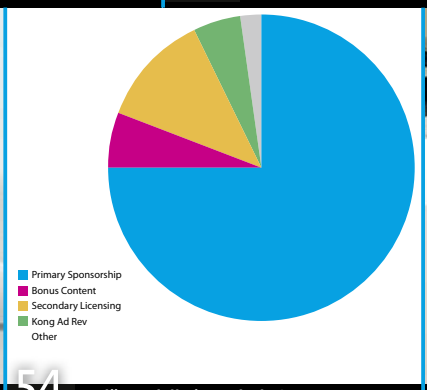


28 Mit nur wenigen Assets eine riesige Grafik-Bibliothek erstellen.

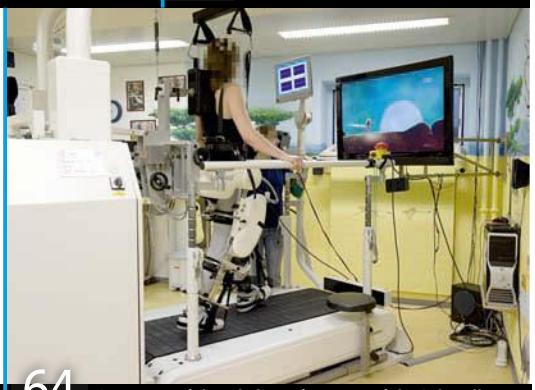
60 Wie Spieler unbewusst gelenkt werden.

48 Aus Dextrose wird Zynga Germany.

20 Wir sprechen mit Rare und Ubisoft Montreal über Kinect.



54 Erlösmodell eines Flash Games.



64 Computerspiel und Therapie: Das Projekt Gabarello.

Entwicklung

32 **Workshop (Serie)**
Von der Idee zum Konzept
von Daniel Dumont

36 **Programmierung**
Parallelisierung für die Xbox 360
von Oliver Iking

40 **Workshop (Serie)**
Testdokumentation: Test Plan & Test Case
von Frank Fay

Business

48 **Interview**
Im Gespräch mit Zynga Germany
von Yassin Chakhchoukh

54 **Post Mortem**
Revenue Shares bei einem Flash Game
von Andy Moore

58 **Interview**
Im Gespräch mit Tau Petersson
von Yassin Chakhchoukh

Campus

60 **Essay**
Beeinflussung von Spielern
von Richard Winterstetter

64 **Forschung**
Gabarello: Ein Roboter als Game Controller
von Cornelius Müller, René Bauer,
Reto Spoerri, Prof. Ulrich Götz, Florian Faller

68 **Ausbildung** Praktikumsbörse

Märkte

71 **Marktforschung**
GameStar- und GamePro-Leserdaten
von Yassin Chakhchoukh

72 **Firmenregister** Die Branche im Überblick

Deutscher Entwicklerpreis 2010

Den Jungs und Mädels von Blue Byte sind mit »Die Siedler« gleich zwei erstaunliche Spagate gelungen: Zum einen haben sie die traditionsreiche Aufbauspiel-Reihe mit dem siebten Teil generalüberholt, ohne dabei die langjährigen Serienfans zu verschrecken. Zum anderen haben sie 2010 nicht nur ein AAA-Siedler veröffentlicht, sondern darüber hinaus noch ein Browser-Siedler sowie ein Facebook-Siedler. Klarer Fall, die fünf Awards – unter anderem für »Bestes deutsches Spiel« und »Bestes deutsches Studio« – gehen völlig zu Recht nach Düsseldorf. Allerdings fehlte es in diesem Jahr auch ein wenig an echter Konkurrenz, bezeichnenderweise gewann mit »Anno 1404: Venedig« dann auch ein Addon die zwei Fachpreise für »Beste Grafik« und »Bestes Game Design«.

Blue Byte war der große Sieger auf der feierlichen Gala in der Essener Lichtburg, aber es gab natürlich noch viele weitere Gewinner – vielleicht sogar zu viele. Insgesamt wurden 25 Awards verliehen. Nicht jedem Zuschauer war dabei etwa klar, wo nun genau der Unterschied zwischen »Bester Kunden-Support« und »Bestes Community Management« lag. Auch das abermals geänderte Regelwerk wurde auf der (erneut famosen) Aftershow-Party heiß diskutiert. Hier gibt es sicherlich Reformbedarf. In einer Sache waren sich jedoch auch in diesem Jahr wieder alle einig: Der Deutsche Entwicklerpreis hat für die Branche eine enorme Bedeutung und wird sie auch weiterhin haben. Im nächsten Jahr dann hoffentlich wieder mit einem stärkeren Konkurrenzfeld.

Die Sieger

Fachpreise

Bestes deutsches Spiel:
Die Siedler 7 (Blue Byte)

Beste Grafik:
Anno 1404: Venedig (Related Designs / Blue Byte)

Beste Story:
A New Beginning (Daedalic Entertainment)

Bestes Game Design:
Anno 1404: Venedig (Related Designs / Blue Byte)

Bester Soundtrack:
A New Beginning (Daedalic Entertainment)

Bestes Studio:
Blue Byte

Bester Publisher:
Ubisoft

Genrepreise

Bestes deutsches Strategiespiel:
Die Siedler 7 (Blue Byte)

Bestes deutsches Adventure:
Lost Horizon (Animation Arts)

Bestes deutsches Rollenspiel:
Drakensang: Am Fluss der Zeit (Radon Labs)

Bestes deutsches Actionspiel:
Trapped Dead (Crenetic)

Bestes deutsches Sportspiel:
Drivals (King Art)

Bestes deutsches Kinderspiel:
Galaxy Racers (kunst-stoff)

Bestes deutsches Jugendspiel:
A New Beginning (Daedalic Entertainment)

Bestes deutsches Familienspiel:
Winter Sports 2011 (49games)

Bestes deutsches Lernspiel:
The Skillz (Daedalic Entertainment)

Publikumspreise

Bestes deutsches Browsergame:
Farmerama (Bigpoint)

Bestes deutsches Social Game:
Die Siedler: Meine Stadt (Blue Byte)

Bestes Mobile Game:
Anno – Erschaffe eine neue Welt (HandyGames)

Bestes Community Management:
Die Siedler 7 (Blue Byte)

Bestes Konsolenspiel:
ArcaniA – Gothic 4 (Spellbound Entertainment)

Bester Kunden-Support:
NosTale (Gameforge)

Bestes Online Game:
Runes of Magic: Chapter III – The Elder Kingdoms (Frogster)

Bester Games Händler:
Amazon

Bestes Games Portal:
Bigpoint.com

Global Game Jam 2011

Mehr als 100 Standorte, 72 Stunden Dauer-Coden, hunderte neue Indie Games: Vom 28. bis zum 30. Januar. werden sich beim Global Game Jam wieder weltweit über 1.000 Designer, Programmier und Grafiker in den Entwicklungs-Marathon stürzen. Deutschland ist mit den Städten München, Bremen, Berlin und Köln am Start.

2009 waren insgesamt 1.600 Teilnehmer in 23 Ländern mit dabei, 370 Spiele wurden damals innerhalb der kurzen Zeit entwickelt – 2010 schoss die Zahl der Teilnehmer bereits auf mehr als 4.000, rund 900 Titel sind in diesem Jahr entstanden.

Die Besonderheit in 2011: Es entstehen nicht nur digitale, sondern auch klassische analoge Spiele. Wer also schon immer mal ein neues Risiko oder Monopoly entwickeln wollte, hat im Januar die Chance. Weitere Infos zu den deutschen Standorten finden Sie unter **globalgamejam.org**

Festival of Games 2011

Am 28. und 29. April findet im niederländischen Utrecht das mittlerweile sechste Festival of Games statt. Mit dem Thema »Games Anywhere« legt die Konferenz den Fokus auf die Medienkonvergenz und trägt damit dem Umstand Rechnung, dass sich Spiele-Entwicklung und Vermarktung über immer mehr mediale Kanäle erstrecken. An den zwei Konferenztagen werden Sessions in den Bereichen »Business and Finance«, »Marketing and Strategy«, Art, Design, Production und Technology abgehalten.

Im Jahr 2010 konnte das Living Games Festival 2.000 Besucher aus 25 Nationen anziehen, zu den Sprechern zählten unter anderem Ian Livingstone, David Perry und der Pac-Man-Erfinder Toru Iwatani. Falls Sie im April dabei sein möchten: Making Games ist offizieller Medienpartner und verlost für das nächste Festival of Games zwei Mal zwei Eintrittskarten im Gesamtwert von fast 1.200 Euro. Schicken Sie einfach eine Mail mit Ihren vollständigen Kontaktdaten an **info@nlgd.nl, Betreff: Making Games**

Ian Livingstone und Toru Iwatani.

G.A.M.E. Bundesverband strukturiert sich neu

Bundesverband der Entwickler von Computerspielen

Der Bundesverband der Entwickler von Computerspielen G.A.M.E. hat sich neu aufgestellt. Mit Funatics-Geschäftsführer Thomas Friedmann wurde auf der Mitgliederversammlung am 2. Dezember ein neuer Vorstandsvorsitzender gewählt, der die Nachfolge von André Blechschmidt antritt. Bigpoint-CEO Heiko Hubertz übernimmt die Rolle des stellvertretenden Vorsitzenden. Der G.A.M.E. richtet sich nun deutlich auf den Online-Markt aus, dies zeigen auch die Berufungen der weiteren Vorstandsmitglieder aus den Unternehmen Gameforge, Upjers und SevenOne Intermedia.

Der Verband will im kommenden Jahr sein Hauptaugenmerk auf den Verbraucher- und Jugendmedienschutz legen, vor allem vor dem Hintergrund des neuen Jugendmedienschutzstaatsvertrages. Darüber hinaus werden 2011 die Mitgliedsbeiträge angepasst, um nach eigener Aussage die Professionalisierung des G.A.M.E. »auf politischer Ebene überzeugend und nachhaltig umzusetzen.« Im Januar wird zudem ein Nachfolger für den derzeitigen Hauptgeschäftsführer Stephan Reichart bestimmt. Dieser hatte den G.A.M.E. in den vergangenen Jahren ehrenamtlich geleitet. Die neue Geschäftsführung wird diese Arbeit ab 2011 hauptberuflich fortführen.

Vier Fragen an ...

Thomas Friedmann

ist CEO von Funatics und Vorstandsvorsitzender des G.A.M.E.

Making Games 4,5 von 5 Vorstandsmitgliedern kommen aus der Onlinegames-Branche. Welche Folgen hat das für die weitere Verbandsarbeit und welche Rolle spielen die Entwickler klassischer Boxed Games?

Thomas Friedmann Diese klassische Definition in »Online« und »Boxed« sehen wir als eher historische Trennung eines Marktes, der in den letzten Jahren zusammen gewachsen ist und weiter zusammen wachsen wird. Ein Großteil unserer Mitgliedsfirmen beim G.A.M.E. ist übrigens bereits heute im oder für das Online-Business tätig. Daher ist es auch nicht ungewöhnlich, dass der neue Vorstand entsprechend geprägt ist. Für uns ist es sogar ein sehr starkes Signal, dass sich erfolgreiche Unternehmen und Marktführer bei uns im Vorstand engagieren, wodurch wir politisch ein noch stärkeres Gewicht bekommen und die Interessen der deutschen Spiele-Industrie noch besser vertreten können.

Making Games Außer der Neuwahl des Vorstands: Was waren die deiner Meinung nach wichtigsten Beschlüsse der Mitgliederversammlung?

Thomas Friedmann Zum einen haben wir nach fast sieben Jahren eine neue Beitragsstruktur verabschieden können, die den G.A.M.E. auch finanziell auf eine solidere Basis stellt und uns eine weitere Professionalisierung erlaubt. Zum anderen haben wir verschiedene Maßnahmen beschlossen, die das gleiche Ziel verfolgen – G.A.M.E. professionalisieren und noch stärker machen. Strukturell und personell. Zu letzterem ist zu sagen, dass sich Stephan Reichart wie angekündigt zum 31. Dezember als Hauptgeschäftsführer von G.A.M.E. zurückziehen und es eine hauptberufliche Geschäftsführung geben wird. Stephan möchte ich an dieser Stelle persönlich und im Namen des ganzen Verbandes für sein enormes Engagement und seine aufopferungsvolle Arbeit danken. Wer weiß, wo G.A.M.E. ohne ihn heute stehen würde.

Making Games Was sind für den Verband im Allgemeinen und für dich im Speziellen die wichtigsten Ziele in den sechs Monaten?

Thomas Friedmann Das ist einerseits ganz klar die strukturelle Professionalisierung des Verbandes mit neuer Geschäftsstelle in Berlin, einer hauptberuflichen Geschäftsführung und diversen organisatorischen Verbesserungen. Für bestimmte Themengebiete werden wir Arbeitsgruppen einrichten, an denen sich unsere Mitglieder beteiligen können. Darüber hinaus werden Verbraucherschutz und Jugendschutz wichtige Themen für uns sein.

Making Games Wie werden die Aufgaben zwischen Vorstand und Geschäftsführung aufgeteilt?

Thomas Friedmann Der Vorstand wird die strategische Richtung und die Ziele des Verbandes vorgeben und deren Umsetzung vorantreiben. Die Geschäftsführung ist für die Umsetzung der Ziele, die Betreuung der Mitglieder, die Öffentlichkeitsarbeit und das »daily political business« zuständig.

Eventkalender

WWW makinggames.de

- Online-Eventkalender mit Suchfunktion und Veranstaltungsinfos

JANUAR 2011

International CES

Ort:	Las Vegas, Convention Center
Termin:	6. bis 9. Januar 2011
Thema:	Fachmesse
Kosten:	75 bis 150 Euro
Homepage:	www.cesweb.org

Mobile Games Forum

Ort:	London, Mandarin Oriental Hyde Park
Termin:	26. bis 27. Januar 2011
Thema:	Fachkonferenz
Kosten:	350 Euro
Homepage:	www.mobilegamesforum.co.uk

Game Forum Germany

Ort:	Hannover, Künstlerhaus
Termin:	27. bis 28. Januar 2011
Thema:	Workshop und Networking-Event
Kosten:	Eintritt frei, Anmeldung erforderlich
Homepage:	www.gameforumgermany.de

Global Game Jam

Ort:	München, Mediadesign Hochschule
Termin:	28. bis 30. Januar 2011
Thema:	kollaborative Spiele-Entwicklung
Kosten:	Eintritt frei, Anmeldung erforderlich
Homepage:	www.ggjmuc.de

FEBRUAR 2011

Casual Connect Europe

Ort:	Congress Center Hamburg
Termin:	8. bis 10. Februar 2011
Thema:	Fachkonferenz
Kosten:	350 bis 425 Euro
Homepage:	www.europe.casualconnect.org

Making Games Powerseminar

Ort:	München, IDG-Medienhaus
Termin:	17. Februar 2011
Thema:	Micropayments & Monetarisierung
Kosten:	349 Euro bis 399 Euro
Homepage:	www.makinggames.de/powerseminar

Making Games Talents München

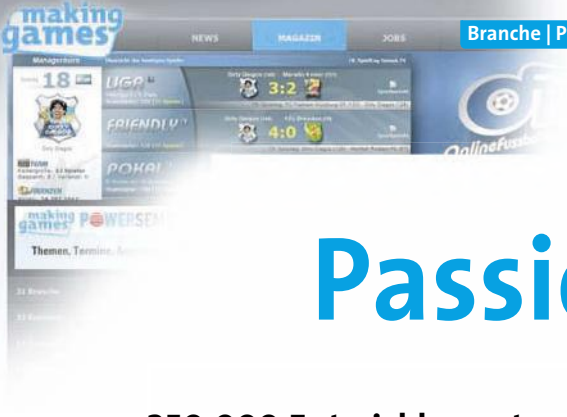
Ort:	München, IDG-Medienhaus
Termin:	19. Februar 2011
Thema:	Recruiting-Event
Kosten:	29 Euro
Homepage:	www.makinggames.de/talents

Game Developers Conference

Ort:	San Francisco, Moscone Center
Termin:	28. Februar bis 4. März 2011
Thema:	Fachkonferenz
Kosten:	55 bis 1.500 Euro
Homepage:	www.gdconf.com

Veranstalten Sie mit!

Sie organisieren eine Podiumsdiskussion oder planen eine Party und möchten, dass die Branche davon erfährt? Dann schreiben Sie uns an **info@makinggames.de**, **Stichwort »Eventkalender«** und wir veröffentlichen Ihre Veranstaltung in unserer nächsten Ausgabe.



Passiert auf making games.de

250.000 Entwickler nutzen die Unity Engine

Innerhalb eines Jahres hat sich die Nutzer-Anzahl der Unity-Entwicklerplattform von 13.000 auf 250.000 vergrößert. Mit der Unity-Engine können Videospiele und 3D-Echtzeit-Animationen für iPhone, iPad, PC, Mac, Android, Wii, Xbox 360 und Playstation 3 erstellt werden. Die Basis-Version kann kostenlos auf der Webseite von Unity Technologies heruntergeladen werden, es existiert aber auch eine kostenpflichtige Pro-Variante.

Für Browser hat die Firma außerdem den Unity Web Player entwickelt, mit dem sich 3D-Inhalte auch über das Internet darstellen lassen. Das Plugin wurde seit der Veröffentlichung über 35 Millionen Mal heruntergeladen und installiert.

Neue Debatte um Hakenkreuze in Spielen

Der Münchner Entwickler Reality Twist möchte in einem pädagogischen Adventure zur deutschen Nachkriegszeit auch Hakenkreuze darstellen. Bisher ist dies in Deutschland undenkbar. Reality Twist hat sich mit seinem Anliegen nun an den FDP-Bundestagsabgeordneten Jimmy Schulz gewandt, der das Thema auf die Tagesordnung setzen möchte.

Produzent Sebastian Grünwald: »In unserem Lernspiel möchten wir natürlich das Zeitgeschehen authentisch dokumentieren. Dabei stellte sich die Frage, wie wir wichtige zeithistorische Ereignisse, wie zum Beispiel die Demontage von Hakenkreuzen, thematisieren sollen, ohne eine Straftat zu begehen.« Das Lernspiel wird vom FilmFernsehFonds Bayern und der Bayerischen Staatskanzlei gefördert und verfolgt konkrete, pädagogische Absichten. »Es kann doch nicht sein, dass TV-Dokumentationen das seit Ewigkeiten problemlos thematisieren dürfen und wir dazu gezwungen sind, die nationalsozialistische Symbolik totzuschweigen.«

Mehr als 50 3D-Spiele für Playstation 3 in Arbeit

Sony will sich auf dem wachsenden Markt der 3D-Konsolenspiele die Marktführerschaft sichern und bereitet ein Portfolio von mehr als 50 Titeln für die Playstation 3 vor. Rund 20 davon entwickelt der japanische Konzern selbst, der Rest wird von externen Studios produziert.

Dies verriet SCEE Senior Director Mick Hocking wenige Tage vor seiner Keynote-Rede zum Branchentreffen Develop in Liverpool: »Wir statten mehr als 20 interne Spiele von Sony mit 3D aus. Insgesamt werden mehr als 50 Titel zu 3D konvertiert und diese Zahl steigt stetig an. Einige davon sind ganz große Namen. Das ist ein wirklich starkes Zeichen dafür, dass die Industrie mit uns übereinstimmt und 3D als ein Kernelement für die Zukunft des Konsolenmarktes sieht.«

Zur stereoskopen 3D-Darstellung benötigt die Playstation 3 neben der aktuellen Firmware auch einen entsprechenden Fernseher und eine 3D-Brille, die unter anderem auch von Sony angeboten werden. Entsprechend optimistisch zeigt sich Mick Hocking für die Zukunft der neuen Technologie: »Spätestens wenn die Leute die ganze Bandbreite an 3D-Inhalten sehen, werden sie sich für ein 3D-TV-Set entscheiden.«

Making Games Links

Making Games ist jetzt auch im Berufsnetzwerk LinkedIn vertreten. Die neue Gruppe für die deutschsprachige Spielebranche.
makinggames.de/linkedin

In diesem Artikel auf Gamasutra wird die crossmediale Vermarktung von Spielen beleuchtet.
makinggames.de/crossmedia

Mit einem Nachtsichtgerät hat ein Xbox-360-Besitzer die Kinect-Strahlen sichtbar gemacht.
makinggames.de/kinectstrahlen

»Wie stellen Sie sich einen typischen Spiele-Entwickler vor?«, fragte der russische Casual Games-Developer Nevosoft Frauen in St. Petersburg.
makinggames.de/klischee

Twisted Pixels (The Maw, Splosion Man) Creative Director Josh Bear erklärt, wie wichtig starke Persönlichkeiten in Videospielen sind.
makinggames.de/personality

Bigpoint-Chef Heiko Hubertz warnt vor Facebook-Abhängigkeit

Heiko Hubertz warnt Spieleentwickler davor, mit ihren Social-Games-Projekten lediglich auf Facebook zu setzen. In der Branche würde der Erfolg zu oft an den Zahlen in Facebook gemessen. Das größte soziale Netzwerk der Welt sei dabei nicht ohne Tücken.

»Es ist wirklich schwierig, sich ganz auf Facebook zu verlassen. Sie können die Regeln ändern, wie es ihnen passt [...]. Das Neueste sind Facebook Credits. Sie können die Leute dazu zwingen, sie zu benutzen aber sie nehmen auch einen 30 Prozent-Anteil. Und wer hat eine 30 Prozent-Marge und möchte sie an Facebook abgeben? Das ist der Grund, weshalb wir uns nicht auf eine Distributionsquelle verlassen und alle möglichen Portale und Plattformen nutzen.« Trotz der Kritik weiß Hubertz auch die Errungenschaften von Facebook zu würdigen: »Das Wachstum von Facebook hat viele Menschen mit Mikrotransaktionen vertraut gemacht. Sie haben dadurch verstanden, wie sie unsere Titel spielen und dafür bezahlen können.«

Atari erwirtschaftet die Hälfte des Umsatzes online

Der Publisher Atari hat die Zahlen zur ersten Hälfte des Geschäftsjahres 2010 bekannt gegeben. Im Berichtszeitraum, der am 30. September endete, nahm das Unternehmen 29,6 Millionen US-Dollar ein – ein deutlicher Rückgang gegenüber dem ersten Halbjahr 2009, als sich die Umsätze auf 68,5 Millionen US-Dollar beliefen. Am meisten litt das klassische Retail-Geschäft. Erwirtschaftete Atari in den ersten beiden Quartalen 2009 noch 66,7 Millionen US-Dollar mir dem ehemaligen Kerngeschäft, so sind es aktuell nur noch 16,7 Millionen. Im Gegensatz dazu hat sich der Umsatz mit Onlinespielen und digitalen Downloads verfielfacht. In diesem Geschäftsfeld konnte der Konzern einen Sprung von 1,8 auf fast 13 Millionen US-Dollar verzeichnen.

making games

POWERSEMINAR

DAS FORTBILDUNGSPROGRAMM FÜR DIE GAMES-BRANCHE

- maximal 20 Teilnehmer
- ausgewiesene Fachleute als Referenten
- acht Stunden fokussierter Workshop
- verblüffende Best-Practice-Beispiele
- Tools für die Praxis

DIE NÄCHSTEN WORKSHOPS

17. Februar 2011
Micropayments & Monetarisierung

Dr. Jarg Temme
CEO deal united

Jürgen Weichert
CSO deal united

Die Schwerpunkte

- Geld verdienen! – Monetarisierung von Usern in Zeiten sich wandelnder Geschäftsmodelle.
- Virtual Goods – die Antwort auf alle Monetarisierungsfragen oder nur der nächste Schritt?
- Ansätze zur Monetarisierung des Funnels.
- Kreditkarte, PayPal! Gibt es noch mehr? Professionelles Micropayment für Ihre virtuelle Währung.

7. April 2011
Virales Marketing & Alternate Reality Games

Thomas Zorbach
Geschäftsführer
vm-people

Die Schwerpunkte

- Alternate Reality Games als virales Marketinginstrument für kleine, mittelständische und große Unternehmen.
- Blogs, Facebook und Twitter – wie man eine Geschichte in den sozialen Medien erzählt.
- Levels of Involvement – quantitative und qualitative Erfolgsmessung.
- Konzeptentwicklung und Erstellen eines ARG-Storydrafts.

Frühbucher sparen!

Nutzen Sie unseren Early-Bird-Tarif und sichern Sie sich das komplette Workshop-Paket für nur **349 Euro** (statt 399 Euro). (inkl. Kongressdokumentation, Zertifikat und Business-Lunch)

Das Seminar ist auf 20 Teilnehmer begrenzt.

jetzt anmelden ➡ makinggames.de/powerseminar

Updates im Profi-Check
Making Games Tools: Unity3D 3.0

Normalerweise bekommen Sie an dieser Stelle mehrere Entwicklermeinungen. Das Update zu Unity3D ist jedoch derart umfangreich und interessant, dass eine ausführliche Expertenanalyse angebracht ist.

Sina Konz

ist Unity3D-Entwicklerin bei Gameforge.

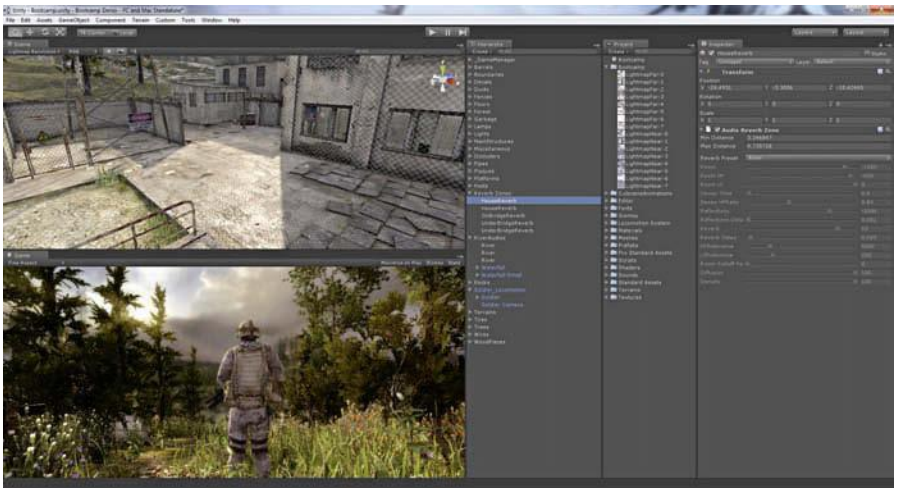
Sina verfolgt ihre Passion für 3D-Entwicklung schon seit ihrem Studium und beobachtet die Weiterentwicklung von Unity3D mit großem Interesse. In der Web-Games-Sparte der Gameforge AG ist sie verantwortlich für die technologische Umsetzung von 3D-Projekten und freut sich darauf, den Fortschritt von 3D im Web zu unterstützen.

Unity 3D ist ein plattformunabhängiges Game Development Tool auf Basis einer mächtigen Game Engine, die auch im Browser grafisch hochwertige 3D-Titel ermöglicht. Im September dieses Jahres veröffentlichte Unity Technologies ein neues großes Update: Unity3D 3.0. Im Folgenden stelle ich Ihnen kurz die wichtigsten Neuerungen vor.

Deferred Rendering

Die wohl größte Neuerung der Engine ist das »Deferred Lighting System«. Im Gegensatz zu herkömmlichen Methoden (Forward Rendering) erfolgt die Beleuchtung hier nicht mehr geometriebasiert, sondern kamerabasiert. Statt die gesamte beleuchtete Geometrie pixelweise abzutasten, werden nur noch die letztendlich sichtbaren und beleuchteten Pixel des Kamerabildes unter Verwendung des G-Buffers abgetastet. Durch diese Methode wird der Berechnungsaufwand weitestgehend von der Anzahl der Lichtquellen losgelöst, was wesentlich besser ausgeleuchtete Szenen ohne großen Performanceverlust ermöglicht.

Im Zuge dieser großen Änderungen wurde auch das bestehende Shader System komplett erneuert. Die neuen Surface Shader sollen die Entwicklung erleichtern, weil der benötigte Source Code wesentlich übersichtlicher und einfacher gehalten ist als im alten System.



Unity3D 3.X bietet zahlreiche neue Funktionen und ermöglicht spektakuläre 3D-Spiele im Browser. Ein Upgrade oder Umstieg ist jedoch nicht bei jedem Projekt zu empfehlen.

Debugging

Das wohl meist ersehnte Technikfeature in Unity3D 3.X ist das neue Debugging Tool. Mithilfe der Entwicklungsumgebung MonoDevelop 2.4 werden nun Funktionen wie das Setzen von Breakpoints, die Inspektion von Variablen und Line-by-Line Single Stepping unterstützt.

Beast Lightmapping

Ein weiteres neues Grafik-Highlight ist die Integration des »Beast Lightmapping Systems«, das bereits in Spielen wie Mirror's Edge oder Mortal Kombat vs. DC Universe eingesetzt wurde. Das Lightmapping System bietet bessere Licht- und Schatteneffekte für In- und Outdoor Settings. Nicht nur die eigentlichen Lichtquellen werden dabei in die resultierende Lightmap eingerechnet, sondern auch die Lichtausstrahlung glühender Objekte, d.h. der Objekte, die einen Shader mit Emission besitzen. Über »Dual Lightmapping« wird je nach Distanz zwischen Lightmap- und Echtzeit-Beleuchtung geblendet, wobei weiterhin das indirekte Licht der Lightmap verwendet wird. Durch verschiedene Einstellungsmöglichkeiten wie zum Beispiel der Blending-Distanz bietet Unity3D wie gewohnt die Möglichkeit, die optimalen Parameter zwischen Qualität und Performance zu finden.

Audiosystem

Mit dem großzügig erweiterten Audiosystem kann sich Unity3D 3.X auch hören lassen. So gibt es diverse Audiofilter und -effekte wie beispielsweise Distortion, Echo, Chorus, Hoch- und Tiefpassfilter. Neu sind auch die Reverb Zones, die zu einem realistischeren Klangerlebnis führen. Wählbare Audioumgebungen wie beispielsweise Konzerthalle, Steinraum, Höhle etc. sorgen für unterschiedliche Halleffekte in den jeweiligen Spielbereichen.

Ähnlich zu den bereits vorhandenen Animationskurven kann man nun verschiedene Komponenten des Audio Rolloffs in Form einer Audio-

kurve definieren. Die Audio Preview macht es möglich, die Sounds direkt im Editor zu testen, und zwar ohne das Spiel zu starten.

Performanz

Zuletzt ist vielleicht noch zu sagen, dass Unity3D auch in dieser Version wieder einige Anstrengungen unternommen hat, die Performanz der Spiele zu verbessern. Diverse Beschleunigungsverfahren wie die Einbindung des Occlusion Culling Systems von Umbra und ein komplett erneuertes Rendering Backend sollen das Spiel um bis zu 50% beschleunigen.

Unity3D 3.X bietet eine Vielzahl weiterer Neuerungen wie beispielsweise die Implementierung einer Socket Security Sandbox, Cloth Simulation, Upgrades der NVIDIA PhysX Library und des Mono und C# Compilers.

Browserintegration

In der alten Version des Unity3D Plugins gab es einige Stabilitätsprobleme des Web Players. So war besonders der Tab-Wechsel innerhalb eines Browserfensters problematisch. Unter Firefox konnte dies zu einer CPU-Auslastung von 100% führen, was ein Einfrieren nach sich zog. Unter Chrome führte ein Tab-Wechsel während des Plugin-Ladens sogar zu einem Plugin-Crash.

Ein unterschiedliches Verhalten des Web Players in Firefox und Chrome ist auch in der Version 3.X festzustellen. So wird das Unity3D-Spiel (bei aktivierter »Run in Backround«-Option) bei einem Tab-Wechsel in Chrome pausiert, die CPU Auslastung fällt auf nahezu 0%. In Firefox läuft das Spiel dagegen weiter, die CPU-Auslastung war jedoch auf den meisten Systemen gleichbleibend, steigt also nicht mehr wie zuvor auf 100% an. Auch der Plugin-Crash unter Chrome scheint weitestgehend behoben.

Vor allem auf Windows-7-Systemen wird man jedoch noch etwas an der Stabilität arbeiten müssen. Unter Chrome führt ein Tab-Wechsel, ähnlich wie zuvor unter Firefox, zu einem Einfrieren des Browserfensters und in ungünstigen Fällen kann es auch hier zu Abstürzen kommen.

Rückwärtskompatibilität

Nichtumsteiger können bei dem Thema Rückwärtskompatibilität ihres Web Games aufatmen. Durch die von Unity3D verwendete Plugin-Kapselung ist diese vollständig garantiert. Anders verhält es sich dagegen im Unity3D-Editor. Besonders aufgrund der einschneidenden Änderungen in der Render Engine wird ein einfaches Upgrade des bestehenden Projekts auf die neue Version fehlschlagen, hier muss man als Entwickler in jedem Fall noch nacharbeiten.

So sind in der neuen Version dynamische Schatten zusammen mit Forward Rendering nur noch für ein Richtungslicht pro Szene möglich, für Point- und Spotlights können sie nicht mehr berechnet werden. Aktiviert man deshalb die Deferred-Rendering-Option, so muss man auf Anti-Aliasing verzichten, da diese Kombination aktuell nicht unterstützt wird.

Durch die Neuerung des Shader Systems müssen Asset Bundles mit Surface Shadern neu erzeugt und alle selbst entwickelten Shader umprogrammiert werden. Durch einen Umstieg auf die neueste NVIDIA PhysX Library muss man eventuell Parameter anpassen, um das gleiche Verhalten wie noch in Version 2.x zu erreichen. Die Implementierung des neuen Security Models erfordert für entsprechende Projekte ebenfalls Anpassungen.

Wenn das entsprechende Projekt der Versionskontrolle unterliegt, sollte es vorher im besten Fall ausgecheckt, auf Unity3D 3.0 angepasst und dann komplett neu eingecheckt werden. Weil Unity3D bei einem Upgrade auch Eingriffe in die Ordnerstruktur des Projekts vornehmen kann, ist sonst eine saubere Version eventuell nicht garantiert.

Fazit

Die flotte Reaktion von Unity Technologies auf neue Technologien hat ihre Vor- und Nachteile. Zum einen wurde auch diese Version wieder auf einen neuen Technikstand gebracht und um hochwertige Features erweitert. Auf der anderen Seite ist bei sehr einschneidenden Änderungen die Rückwärtskompatibilität des Editors nicht immer gewährleistet.

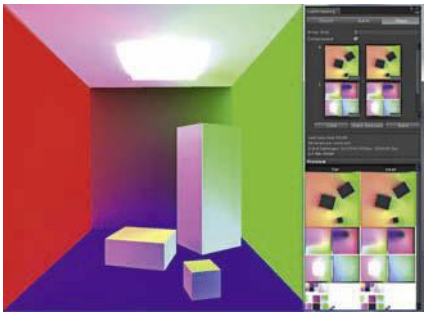
Für ein Update dieser Größe hätte ich mir, vor allem im Hinblick auf die fehlende Rückwärtskompatibilität, mehr Informationen vonseiten Unity Technologies gewünscht, zum Beispiel eine detaillierte Checkliste mit Änderungen, die man bei einem Umstieg vornehmen muss. Das würde die teils mühevolle Fehlersuche und Recherche um einiges erleichtern.

Ein Nachteil von Unity3D 3.X ist außerdem die fehlende Anti-Aliasing-Unterstützung des Deferred Rendering, besonders weil in dieser Version einige Funktionen stark an diese Technik gekoppelt sind. Forward Rendering unterstützt zwar weiterhin Anti-Aliasing, kann aber (im Gegensatz zu Deferred Rendering) in der neuen Version keine dynamischen Schatten mehr für Spot- oder Punktlichter darstellen. Auch die »Dual Lightmapping«-Technik kann nur zusammen mit Deferred Rendering verwendet werden.

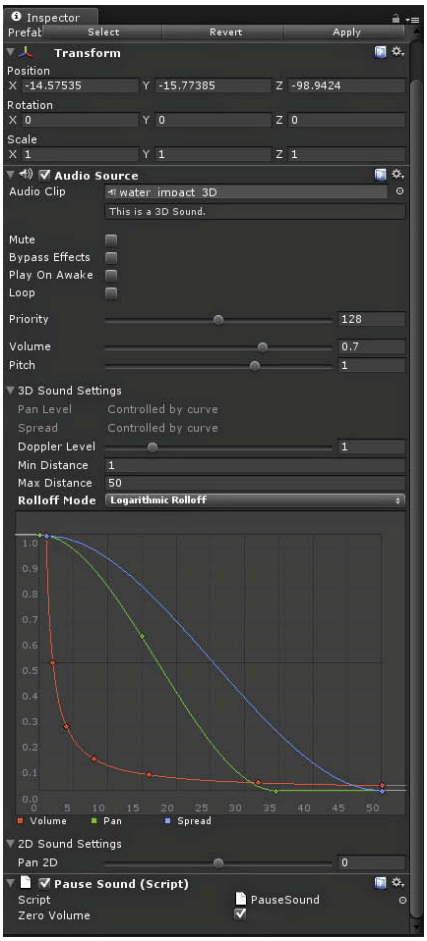
Für wen lohnt sich das Upgrade?

Eine Empfehlung zum Umstieg ist daher vor allem von der Projektkomplexität und der jeweiligen Projektphase abhängig zu machen. Ist das Projekt relativ weit entwickelt oder gar kurz vor einer QA-Phase, ist davon abzuraten. Es ist ein sehr großes Update, weswegen das Projekt nach einem Umstieg sorgfältig auf mögliche, neu aufgetretene Fehler getestet werden muss. Sehr empfehlenswert ist der Umstieg auf Unity3D 3.X dagegen für alle Projekte, die sich noch in der Anfangsphase befinden, zum Beispiel in der Pre-Production oder zu Beginn der Production-Phase. Diese Projekte werden in jedem Fall von dem Mehrwert der neuen Features und Technologien profitieren.

Sina Konz



Das neue **Beast Lightmapping System** kennt man bereits aus AAA-Produktionen wie Mirror's Edge oder Mortal Kombat vs. DC Universe.



Highlights des neuen **Audio Systems** sind die vielen neuen Filter und Effekte wie Distortion, Echo, Chorus, Hoch- und Tiefpassfilter.

Was ist dran am neuen 3D-Hype?
Stereoskopie für Spiele-Entwickler

Stereoskopie in Spielen ist eine komplexe Angelegenheit, die Balance zwischen beeindruckender Tiefenwirkung und störungsfreiem Gameplay eine schwierige Herausforderung. Alexander Taube erklärt, was man über die zukunftsweisende 3D-Technik wissen muss und mit welchen Tricks sie in der CryEngine 3 zum Einsatz kommt.

Alexander Taube

ist Project Lead bei Crytek.

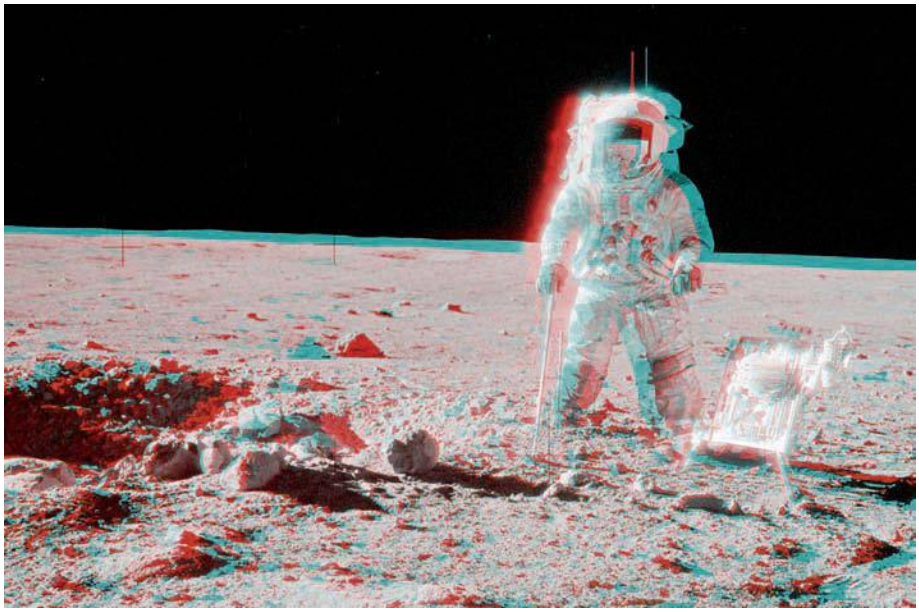
Alexander leitet bei Crytek ein Entwicklerteam, das sich mit der Erforschung innovativer Spielkonzepte befasst. Ein besonderer Fokus liegt dabei auf stereoskopischem 3D in Spielen. Nach seinem Studium der Psycholinguistik hat Alexander seine Karriere 1994 als Media Designer bei Bavaria Film begonnen und sich dort auf Story Writing, Regie und den Schnitt konzentriert. Später war er für das Jugendförderungsprogramm des Unternehmens verantwortlich. 2001 startete er bei Crytek als Level und Game Designer und arbeitete seitdem an FarCry und Crysis. Neben seiner Tätigkeit als Leiter eines Entwicklungsteams lehrt er derzeit Game Design an der University of Applied Sciences in Darmstadt/Dieburg.

Das Jahr 2009 hat mit »Avatar« eine neue Ära des 3D-Booms eingeläutet. Der Kinofilm hat an der Kinokasse höchst eindrucksvoll bewiesen, wie viel Geld man mit 3D machen kann. Ob letzten Endes wirklich nur der 3D-Effekt dafür verantwortlich war, sei dahingestellt. Fakt ist, dass nun viele Branchen die Chance sehen, mit der 3D-Technologie Kasse zu machen. Ist das neue 3D nur ein kurzer Marketinghype wie die berüchtigte »neue Rezeptur« oder tatsächlich ein echter Zugewinn? Und lässt sich der Kinoerfolg auch auf andere Medien übertragen, eben gerade auf Games?

Die Technik dahinter

Als Technologie ist »3D« schon ein alter Hut und hat bereits vor mehr als 100 Jahren als Stereoskopie die Menschen begeistert. Erste Filme in 3D sind ebenfalls deutlich älter als die Welle, die in den 50er/60er-Jahren startete. Vor rund 60

Das **HUD** war bei der 3D-Entwicklung eine Herausforderung, weil es normalerweise als Fläche auf einer einzigen Ebene liegt und im S3D-Raum deplatziert wirkt. Im Bild ist das HUD zu sehen, wie es nun in Crysis 2 zum Einsatz kommt.



Das **anaglyphe Verfahren** ist vergleichsweise günstig und als einziges auch im Printbereich einsetzbar. Allerdings leidet die Bildqualität durch den Verlust wichtiger Farbinformationen.

Jahren sah man in den dreidimensionalen Filmen eine Chance, sich vom aufkommenden Fernsehen abzusetzen. Leider war die Technik zu aufwendig und unpräzise, die Filme zu effektgeladen, um ein anhaltendes Genre zu stützen. Als Jahrmarktattraktion (in Form von 4D, 5D und auch 6D ...) haben sich diverse stereoskopische Filme zwar irgendwo gehalten, aber die Marktanteile fielen bescheiden aus.

Woher kommt jetzt der Wandel? Ganz klar, das digitale Zeitalter macht's möglich. Die CGI-Filme haben es vorgemacht, dass moderne Digital-Kinoprojektoren zwar teuer in der Anschaffung sind, aber dem Kino (mit einem Adapter) ermöglichen, vergleichsweise günstig die Investition in die digitale Technik in einen echten Mehrwert zu verwandeln. Erst die weite Verbreitung von digitalen Projektoren plus Zubehör hat den Erfolg von Avatar ermöglicht. Also der richtige Film zur richtigen Zeit, um ein klares Statement zu setzen: 3D ist angekommen und hat vor, auch zu bleiben.

Ein verbleibender Kraftakt stellt für das Kino allerdings noch die Produktion von realen Filmszenen in 3D dar. Hier ist der Aufwand immer noch deutlich höher, weshalb wir in naher Zukunft auch weiterhin mehr CGI-3D-Filme sehen werden als Nicht-CGI-Filme. Am Rande sei erwähnt, dass einige Studios ihre Filme auch aufwendig nachträglich »dreidimensionalisieren« lassen, was aber zu einer fragwürdigen Qualität führt.

Games jetzt auch in 3D?

Games in 3D sind eigentlich nichts Neues. Spiele sind spätestens seit dem Erfolg von Doom oder Quake in den 90er-Jahren in der Lage, in 3D zu laufen. Der große Unterschied ist hier die Präsentation für den Spieler – bisher wurden uns diese Spiele in 2D geliefert. Ein klassischer PC-Monitor oder Fernseher erlaubt uns nur einen zweidimensionalen Blick in eine dreidimensionale Welt (**siehe Kasten »3D« ungleich »S3D«**).

Der Vorteil für Spiele-Entwickler ist hier, dass die technische Umstellung der Produktion eines »3D«-Games gar nicht so kompliziert ist. Einige Punkte muss man aber doch beachten, wie weiter unten beschrieben.

Für die Spielebranche hat der »Umstieg« von 3D auf stereoskopes 3D (S3D) ähnliche Konsequenzen wie im CGI-Filmbereich. Das Grundmaterial liegt bereits digital vor und ist grundsätzlich für den Einsatz einer stereoskopischen Darstellung vorbereitet. Die erste Hürde ist es, einen Überblick über die unterschiedlichen Möglichkeiten des S3D zu gewinnen.

Moderne 3D-Hardware

Wenn man die gesamte technologische Kette betrachtet, benötigt zunächst einmal der Spieler eine Plattform, die S3D-fähig ist (PC, Konsole). Hinzu kommt ein Ausgabegerät (Monitor, Fernseher), das S3D darstellen kann. Weil viele unterschiedliche Standards existieren, müssen beide Einheiten auch noch miteinander kommunizieren können.

Die gängigen Darstellungsverfahren

Das Grundprinzip von S3D liegt darin, jedem Auge ein getrenntes Bild zu liefern, ganz wie im »realen« Leben auch – jedes Auge hat eine leicht verschobene Perspektive. Für einen 3D-Effekt benötigt man also ein Bild, das eigentlich zwei Bilder beinhaltet, die wiederum durch einen Filter oder durch ein Verfahren vor dem Auftreffen auf der Netzhaut getrennt bzw. verdeckt werden.

Anaglyphen

Dieses Verfahren wird den meisten schon irgendwo einmal begegnet sein. Man erkennt es

3D ungleich S3D

Um die Begriffe eindeutig zu verwenden, folge ich einer Terminologie, die sich langsam unter Entwicklern und Spielern durchsetzt. Schon lange werden Games als 3D bezeichnet, wenn sie eine dreidimensionale Welt anbieten, auch ohne eine stereoskopische Präsentation. Wenn man hingegen von der stereoskopischen Darstellung eines Spiels spricht, sollte man den Begriff S3D (Stereoskopes 3D) verwenden.

Die **Shutter-Lösung von Nvidia** arbeitet unabhängig vom Spiel und gibt die Daten von der Grafikkarte direkt stereoskopisch an den Monitor weiter.

an den typischen »bunten« Brillen, von denen ein Glas meist rot, das andere grün oder cyan ist. Auch andere Farbkombinationen sind möglich, diese haben sich aufgrund optischer Nachteile aber nicht durchsetzen können. Durch die Farbfilter werden jedem Auge nur Teile der Bildinformation übermittelt.

Dieses Verfahren ist günstig herzustellen und zurzeit das einzige, das auch im Printbereich eingesetzt oder auf Geräten dargestellt werden kann, die für andere Verfahren nicht speziell vorbereitet sind. So kann etwa jeder PC-Monitor oder TV anaglyphe Bilder anzeigen. Die Qualität ist leider beträchtlich eingeschränkt, weil bestimmte Farbinformationen verloren gehen, die vor allem für die Tiefenwirkung wichtig sind.

Ohne Brille sind die farblichen Abweichungen deutlich zu erkennen. Mit Brille ergibt sich ein überzeugender räumlicher Eindruck.

Erwähnt sei an dieser Stelle eine Verfeinerung des Verfahrens, genannt »Infitec«. Dabei werden nicht zwei Farben als Bildtrennung genutzt, sondern einzelne Bereiche des Lichtspektrums. Dadurch kann jedem Auge ein Anteil des roten, grünen und blauen Spektrums zugeordnet werden. Jedes Auge sieht also ein nahezu vollständiges Farbbild; ein Vergleich mit den klassischen Anaglyphen ist deswegen praktisch falsch, weil das Infitec-Verfahren den Farbfehler behebt. Derzeit ist das System meines Wissens auf Projektionssysteme beschränkt.

Active Shutter

Mit dem Begriff des aktiven Shutters werden solche 3D-Systeme bezeichnet, die aktiv eine LCD-Brille ansteuern. Diese schließt alternierend ein Auge und öffnet parallel das andere, um ebenfalls alternierende Bilder am Monitor oder über Projektionen korrekt an das entsprechende Auge zu liefern.

Die derzeit erhältlichen LCD-TVs nutzen mehrheitlich diese Technologie. Der große Vorteil ist die gleichbleibende Auflösung, jedoch nimmt die Anzahl der Bilder pro Auge um die Hälfte ab, da jedes Bild nur von einem Auge gesehen wird. Aus diesem Grund laufen die Geräte alle mit 100 oder 120 Hertz, was halbiert auf die klassischen Bildwiederholraten der PAL-, SECAM- und NTSC-Sendeformate von 50 bzw. 60 Hertz zurückzuführen ist. Der Nachteil ist die zwingende Nutzung der aufwendigen Brillen, die verhältnismäßig teuer sind, relativ schwer und natürlich auch – da kabellos – irgendwann wieder geladen oder getauscht werden müssen (mittels Einwegbatterie oder wieder aufladbaren Akkus). Im Gegensatz zu den Brillen sind die TV-Geräte selbst inzwischen preislich in einem Bereich angekommen, in dem sich die Investition für eine breitere Masse lohnt.

Im Bereich der PC-Monitore sieht es ähnlich aus. Hier dominiert eine Lösung von Nvidia, die mit 120 Hz-Monitoren arbeitet. Diese Lösung arbeitet unabhängig vom Spiel und greift direkt auf die Daten in der Grafikkarte zu, um sie stereoskopisch an den Monitor weiterzugeben.

Passive Polarisation

Ein sogenannter »passiver« 3D-TV nutzt die Eigenschaften des Lichts, über einen polarisierten Filter geblockt werden zu können. Wenn ein Fernseher mit einem solchen Filter ausgestattet ist, wird meist jede Pixelzeile des Bildes gegenläufig polarisiert. Ohne Brille ist dies nicht zu sehen. Mit entsprechender Brille wird jedem Auge nur jeder zweite Pixel angezeigt. Der offensichtliche Nachteil ist die dadurch halbierte Auflösung. Dafür ist die Brille sehr viel leichter und günstiger (keine Elektronik), der Fernseher selbst aber noch teuer in der Herstellung. Zurzeit werden Geräte deshalb lediglich in kleinen Stückzahlen angeboten, hauptsächlich im Bereich Business und Präsentation.

Autostereoskopisch

Als »Technologie der Zukunft« werden die brillenlosen TV-Geräte betitelt, die die Trennung der Bilder für beide Augen bereits im Gerät selbst erledigen. Dabei werden meist die Spalten des Bildes mit kleinen Linsen versehen, die das Licht in einem geringfügig veränderten Winkel aussenden. Dadurch ergeben sich beim Betrachter zwei unterschiedliche Bilder.

Im Augenblick ist diese Technik noch zu teuer für den Massenmarkt. Hinzu kommt bei ihr die Notwendigkeit einer höheren Auflösung, weil ähnlich wie bei der Polarisation jeder Pixel nur von einem Auge gesehen wird. Teilweise verwenden die Geräte bis zu neun (häufiger sind fünf) verschiedene Bilder. Ohne spezielle Anpassung im Fernseher muss auch tatsächlich diese Anzahl an separaten Bildern geliefert werden. Neueste Entwicklungen gehen aber in die Richtung, die Bilder, die der Fernseher darstellt, aus einem einzigen stereoskopischen Paar im Bildschirm zu generieren. Dadurch können auch diese Fernseher mit den üblichen S3D-Bildformaten beliefert werden.

Ohne an dieser Stelle die Feinheiten der einzelnen Verfahren zu erläutern, ist schon jetzt abzusehen, dass die Vielfalt der Systeme immens ist. Leider wird die Situation mit den etablierten Bildformaten nicht besser.

Die verbreiteten Bildformate

Interlaced

Mit Interlaced bezeichnet man ein 3D-Bildformat, das beide Bilder entweder horizontal oder vertikal in einzelne Linien verpackt. Dabei ist jede Linie abwechselnd für das linke und das rechte Auge bestimmt. Die Wurzeln dieses Verfahrens liegen in der klassischen Fernsehsendetechnik, bei der zum Beispiel im PAL-System ein vollständiges Bild aus zwei Halbbildern aufgebaut wird. Diese bestehen wiederum aus der Hälfte der Zeilen, die normalerweise ein vollständiges Bild ausmachen.

Side by Side (up and below)

Das »Side-by-Side«-Verfahren erinnert an die ersten Tage der Stereoskopie, weil beide Bilder nebeneinander dargestellt werden. Weil man

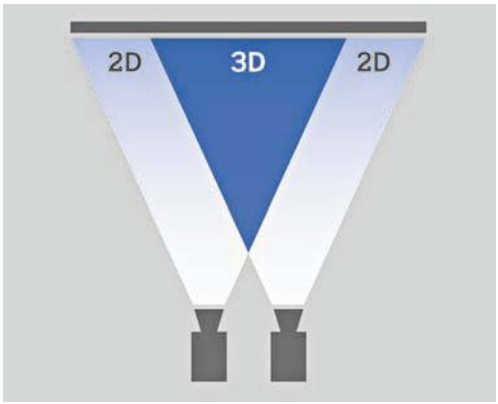


Abbildung 1: Zwei starre Kameras blicken in **dieselbe Richtung**, erzeugen aber nur im Zentrum 3D.

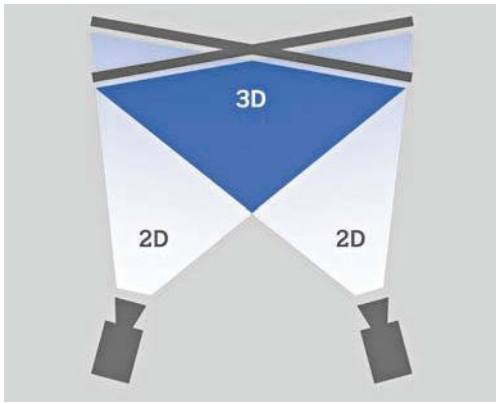


Abbildung 2: Zwei Kameras sind **angewinkelt ausgerichtet**, sodass sie einen gemeinsamen Fokus haben.

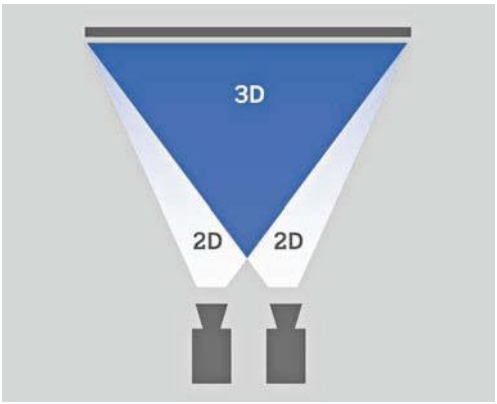


Abbildung 3: Zwei parallele Kameras berechnen ein jeweils **»verzerrtes« Einzelbild** mit gemeinsamer Fokusebene.

sich nach der Auflösung und dem Seitenverhältnis eines einzelnen Bildes richtet, ist das 3D-Bild meist verzerrt, es wird horizontal gestaucht und damit in der Auflösung für ein Auge halbiert. Dieser Kompromiss wird eingegangen, um das zu übertragende Bild über alle Geräte, die das Bild zurücklegt, intakt zu halten. So kann etwa im HD-Format ein Bild in der Auflösung 1920x1080 Pixel @60Hertz übertragen werden. Dabei ist die linke Hälfte des Bildes (960x1080 Pixel) für das linke Auge und die rechte Hälfte für das rechte Auge bestimmt.

Eine nicht ganz so verbreitete Form ist die Übertragung als vertikal gesplittetes Bild. Das technische Prinzip ist zwar dasselbe wie beim horizontalen Splitting, allerdings wird bei diesem Vorgang die vertikale Auflösung halbiert.

Checkerboard

Wie der bildliche Name – das Schachbrett – verdeutlicht, werden hier beide Einzelbilder blockweise verschachtelt. Grundsätzlich ist dieses Verfahren dem Side-by-Side ähnlich, lediglich der Bildaufbau ist unterschiedlich. Ein Vorteil des Checkerboard-Layouts ist die bessere Lesbarkeit von Text. Dieses Verfahren wird aber selten benutzt und ist bei Programmierern – vorsichtig ausgedrückt – nicht sonderlich beliebt.

Ein Nachteil ist die zwingende Notwendigkeit, die exakte Auflösung zu liefern, die das Display nativ unterstützt, weil sonst der interne Upscaler die kodierten Einzelbilder korrumpiert.

Alternierend

Unter diesen Begriff fallen alle Verfahren, die die Bilder der einzelnen Augen in voller Auflösung, aber hintereinander übertragen. Wenn die Bildrate nicht erhöht wird, erkauft man die höhere Auflösung allerdings mit einer halbierten Bildrate. Speziell bei Systemen, die eine Bildrate von 100/120 Hertz erlauben, ist dieses Verfahren beliebt, weil das Fernsehgerät dann keine teuren Anpassungen benötigt.

Weitere Formate

Da es (fast) noch keinen etablierten Standard im TV-Bereich gibt, der die unterschiedlichen Formate vereinheitlicht, können wir erwarten, dass es vielleicht noch mehr werden. Eine große Chance wird dem Anfang 2010 veröffentlichten neuen Standard HDMI 1.4a eingeräumt, der mittlerweile eine breite Palette an 3D-Übertragungsstandards definiert. Unter anderem auch solche, die es erlauben, beide Einzelbilder in voller Auflösung zu übertragen. Dies garantiert die maximale Qualität, benötigt aber auch Geräte, die die neuen Spezifikationen verstehen.

3D in Software

Neben den Formaten von Hardware und Standards spielt auch die Aufbereitung des 3D-Bildmaterials eine entscheidende Rolle. Ausgehend vom Film-Bereich gibt es die zwei folgenden gebräuchlichen Kameraverfahren, um Bewegtbilder in 3D zu erfassen:

Parallele Kameras

Wie dem Schaubild aus **Abbildung 1** zu entnehmen ist, werden bei diesem Verfahren zwei Kameras exakt parallel aufgestellt. Das Ergebnis ist zwar 3D, doch leider nicht in allen Bereichen. Es ergeben sich am Rand vertikale Streifen, die nur von einer Kamera erfasst werden und damit letzten Endes nur »2D« sind.

Zwar sind auch die menschlichen Augen ähnlich angeordnet, wir können unsere Augen aber konvergieren lassen, sprich beide Augen können einen gemeinsamen Punkt fokussieren, was bei diesem Kameraset nicht vorgesehen ist.

»Toe-in«-Kameras

Beim zweiten Verfahren, im Englischen »toe-in« genannt, wird das Verhalten des menschlichen Auges nachgeahmt, einen einzelnen Punkt zu fokussieren (**Abbildung 2**). Das Resultat ist zwar, dass der 3D-Bereich größer ist, dafür wird das Bild zum Rand hin stärker verzerrt. Je nach Brennweite ist die Verzerrung so groß, dass die Bilder nachbearbeitet werden müssen, damit auch wirklich beide Augen die korrekten Flächen als ein Objekt interpretieren.

Ohne nachträgliche Korrekturen lassen sich echte Szenen in 3D also nur eingeschränkt filmen. Zum Glück entstehen unsere Bilder virtu-

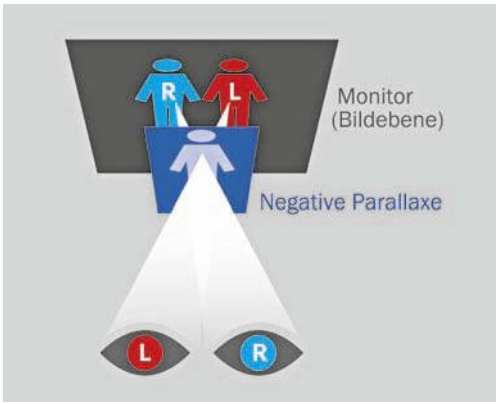


Abbildung 4: Das Objekt befindet sich **vor der Bildebene**. Die beiden Einzelbilder sind quasi vertauscht, wodurch sich die Augen »kreuzen« und das projizierte Bild nach vorn schieben.

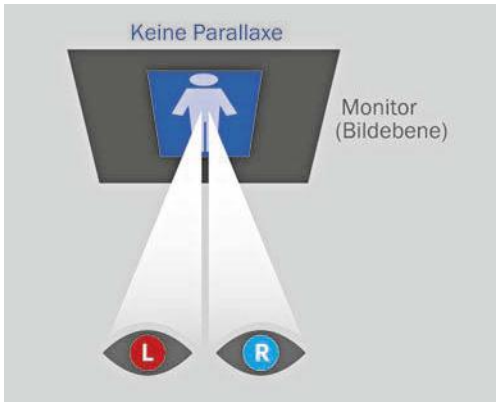


Abbildung 5: Das Objekt ist **in der Bildebene** – beide Einzelbilder liegen aufeinander und haben keine Parallaxe.

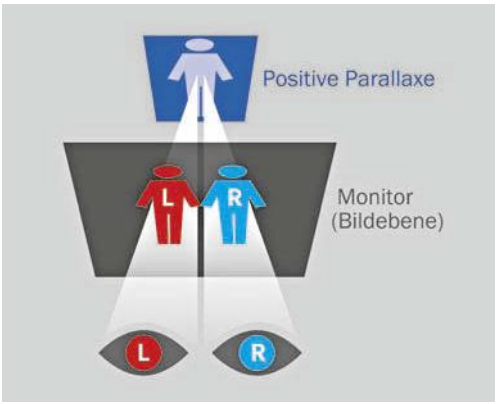


Abbildung 6: Das Objekt befindet sich **hinter der Bildebene**, beide Einzelbilder verschieben sich positiv voneinander weg. Der Abstand zueinander definiert die räumliche Tiefe.

ell im Rechner. Game Developer können daher ein bisschen einfacher tricksen, als dies bei anderen Verfahren möglich ist.

»Frustum Shift«-Verfahren

Wie lassen sich also die Vorteile beider oben erwähnten Systeme ergänzen, ohne neue Nachteile einzuschleppen? In der CryEngine 3 haben wir ein Verfahren eingesetzt, das Kameras ebenfalls parallel zueinander aufstellt, damit die Abbildungsebenen beider Kameras auf der gesamten Fläche liegen. Gleichzeitig wird aber der Frustum verschoben. Einfach erklärt, wird das Zentrum eines Bildes vom Zentrum der optischen Achse verschoben. In **Abbildung 3** ist deutlich zu erkennen, dass beide Kameras nicht im optischen Zentrum stehen, sehr wohl aber ein gemeinsames Bildzentrum teilen. Dadurch, dass die nicht vermeidbare Verzerrung bei der Generierung der Bilder kompensiert wird, bilden beide Bilder exakt den gleichen Bereich ab. Dieses Verfahren ist prädestiniert für eine Implementierung im Games-Bereich oder auch für CGI. Eine Umsetzung für ein Kamerasystem zum Filmen realer Szenen ist mir nicht bekannt (in der Theorie aber durchaus denkbar).

Für Menüs benutzt die CryEngine 3 **Scaleform Gfx** als Middleware, um Flash-Objekte in 3D zu verarbeiten.

Die ersten 3D-Tests wurden mit einem **unveränderten** Crysis durchgeführt.

Die CryEngine 3 in 3D

So weit die Theorie. Wie hat sich die Umsetzung nun aber tatsächlich abgespielt? Um das Thema S3D richtig anzupacken, haben wir mit unserem Team ein experimentelles Spiel entwickelt, das von Anfang an als ein rein stereoskopischer 3D-Titel geplant wurde – eine parallele Spielbarkeit unter 2D-Bedingungen haben wir dabei ganz bewusst ausgeschlossen, um sich ausschließlich auf die kommenden Herausforderungen zu konzentrieren. Dieser Weg klingt recht aufwendig, ist aber auf lange Sicht für uns der richtige gewesen, weil die Erkenntnisse direkt in die CryEngine eingeflossen sind. Wer eine langfristige Nutzung anstrebt, sollte über einen simplen 3D-Patch hinaus denken.

Erste Tests wurden ganz simpel mit einem unveränderten Crysis auf einem 3D-Monitor durchgeführt, der mithilfe eines Treibers dem Rechner ein Stereobild entlockt hat. Schon zu diesem Zeitpunkt war zu erkennen, dass die räumliche Wirkung dem Spiel noch einmal mehr Atmosphäre verlieh – aber wir haben auch gemerkt, dass sich einige implementierte Effekte nicht eins zu eins auf 3D übertragen ließen, wenn diese nicht noch einmal direkt in der Engine angepasst wurden.

Der erste Eindruck

Mithilfe unserer »Research & Development«-Abteilung konnten wir auf einem vorhandenen 3D-Monitor schnell die ersten Resultate betrachten. Intern hat der Renderer zur normalen Kamera eine weitere Kamera gerendert. Trotz diverser kleiner grafischer Fehler war das Ergebnis schon ansehnlich und hat das Potenzial für ein großartiges Spielerlebnis gezeigt.

Die »kleinen« grafischen Fehler haben dann eine Überarbeitung in der Engine nach sich gezogen. So wurden zum Beispiel Schatten nicht korrekt für beide Augen berechnet. Daher mussten noch diverse Algorithmen an die zweite Kamera angepasst werden.

Das größere technische Problem stellte aber die Performance dar. Auf dem PC sind es die Spieler gewohnt, selbst an diversen Einstel-

lungen drehen zu können, um den eigenen Rechner auf die Gegebenheiten der Hardware und Software anzupassen. Es ist aber nicht zu vermeiden, dass die allgemeine Performance für S3D unter diesen Umständen immer auf Kosten der Qualität gehen wird. Für das experimentelle Game spielte das zwar erst einmal nur eine untergeordnete Rolle. Es war aber klar, dass auf lange Sicht eine andere Lösung für das Problem gefunden werden musste.

Eine weitere Herausforderung stellte das HUD dar, also die grafischen Elemente wie Statusanzeigen, Fadenkreuz, Objektmarkierungen und auch Menüs. Diese waren alle in 2D gerendert, selbst wenn sie einen 3D-Look hatten. Über der S3D-Darstellung liegen diese Elemente alle in der gleichen Tiefenebene, nämlich der des Monitors. Für Menüs ist dies erst einmal kein technisches Problem. Es fühlt sich aber für ein S3D-Game deplatziert an, weil das gesamte Menü in einer einzigen Raumtiefe dargestellt werden sollte. Kaum etwas fällt in einem S3D-Spiel mehr auf als das plötzliche Fehlen von Tiefe! Dazu benutzt die CryEngine 3 eine Middleware namens Scaleform Gfx, die Flash-Objekte in 3D verarbeiten kann.

Es sei an dieser Stelle erwähnt, dass gerade die Entwickler, die sich mit dem Stereo-Code beschäftigt haben, häufiger von ihren Augen genarrt wurden – »Ist das jetzt Stereo oder nicht …?« Denn häufig können die Augen eine Menge Fehler kompensieren, ja manchmal sogar 3D vortäuschen, obwohl die beiden Bilder links/rechts-vertauscht sind. So ist es mehr als einmal passiert, dass wir uns eine neue Version anschauten und nur sehr langsam merkten, dass irgendwas nicht stimmte. Ein Drehen der Brille öffnete im wahrsten Sinne des Wortes die Augen – mal wieder beide Bilder vertauscht.

Die Tiefenproblematik

HUD-Elemente lassen sich in zwei Gruppen unterscheiden: in solche, die Informationen mitteilen, die unabhängig von ihrer 3D-Position sind, und in Elemente, die ihre Information in direktem Zusammenhang auf Objekte in der Spielwelt beziehen. Erstgenannte lassen sich gemeinsam in einer Tiefe darstellen bzw. unabhängig vom sonstigen Spielgeschehen.

Grundsätzlich lassen sich Objekte grob in drei Tiefen einteilen: vor dem Monitor (also mit negativer Parallaxe, **Abbildung 4**), genau auf der Projektionsebene (**Abbildung 5**) und in der Tiefe des Raumes, also hinter dem Monitor (mit positiver Parallaxe, **Abbildung 6**). In den meisten S3D-Anwendungen werden die handlungsrelevanten Teile hauptsächlich hinter der Projektionsfläche platziert, weil dies dem gewohnten Sehen am nächsten kommt. Alles, was vor der Projektionsfläche liegt, kann den 3D-Effekt zwar deutlich verstärken, bringt aber Probleme mit sich (**siehe Kasten »3D in your Face!«**).

Welche Tiefe man für die klassischen HUD-Elemente wählt, ist ein bisschen vom Geschmack und vom Stil des Spiels abhängig. Mit zahlreichen Tests haben wir versucht, eine geeignete Position vor dem Monitor zu finden, um eine realistische Wahrnehmung eines HUD, das direkt vor den Augen schwebt, nachzuempfinden. Auch wenn dies einen guten Effekt ergibt, haben wir uns letztlich dagegen entschieden. Anstatt sich harmonisch in das Spiel einzufügen und es zu unterstützen, wurde die Darstellung als zu aufdringlich empfunden. Zusätzlich müssen die Augen des Betrachters deutlich mehr »arbeiten«, um den Unterschied in der Fokussierung zu leisten, was eine schnellere Ermüdung nach sich zieht. Wenn man das HUD näher an der Fokusebene des eigentlichen Spielgeschehens positioniert, wirkt der Gesamteindruck des Spiels deutlich runder.

Objektanpassungen für 3D

Die zweite Gruppe der HUD-Elemente, im Raum platzierte Objekte, wirkt vollkommen unnatürlich, wenn man ihre Position in der Tiefe nicht korrigiert. Verlegt man diese HUD-Elemente näher an die korrekte Position, entfernen sie sich natürlich von der Spielerposition und müssen entsprechend vergrößert bzw. ihre Größe unabhängig von ihrer räumlichen Entfernung gerendert werden. Ein Kompromiss ist, die Größe in Abhängigkeit zu ihrer Entfernung zu variieren, sie aber nicht unter einen bestimmten Wert fallen zu lassen, damit das Element noch zu sehen ist. Diese HUD-Elemente werden auch nicht an der tatsächlichen Tiefenposition des Objekts platziert, sondern nur räumlich angepasst, um ihre Relation in der Tiefe zu unterstreichen. Tatsächlich liest ein Mensch die Position in der Tiefe nur zu einer Entfernung von vielleicht 50 Metern. Alles dahinter wird nicht räumlich in der Tiefe erkannt, sondern durch unser Verständnis von Größe und Relation zu anderen Objekten korrekt in der Tiefe eingeordnet.

Probleme tauchen dann auf, wenn etwa Objekte in der Spielwelt markiert werden, die Markierung korrekt im Raum dargestellt wird, aber das eigentliche Objekt plötzlich hinter einem anderen (z.B. einer Mauer) verschwindet. Wenn man nun die Markierung tiefenkorrekt darstellt, liegt diese im räumlichen Sehen hinter der Mauer, sie durchdringt sie auf unnatürliche Weise. Eine Option ist es, die HUD-Markierung dann nicht darzustellen – doch häufig ist das Anzeigen eines Objekts hinter einer Mauer genau der Grund, warum es eine solche Markierung gibt. Spätestens hier muss man sich im Design entscheiden, welchen Weg man einschlagen möchte. Eine perfekte Lösung, die für alle Spiele gilt, gibt es wohl nicht. In Crysis 2 wird zum Beispiel die 3D-Position des Fadenkreuzes dynamisch angepasst, sodass es immer vor demjenigen Objekt zu schweben scheint, das dem Spieler am nächsten ist. Auch bei der Umstellung der HUD-Objekte von 2D-Grafiken auf 3D-Objekte existiert kein Königsweg – inwieweit hier ein optischer Zugewinn mit der dezenten Darstellung von Informationen harmoniert, muss ein Entwicklungsteam von Fall zu Fall entscheiden.

Die Performance-Tricks der CryEngine

Sobald diese Hürden genommen waren, ging es wieder um das Thema Nr. 1 – die Performance, vor allem auf den Konsolen, die unsere Engine ebenfalls unterstützen muss. Es war keine Option, für die Xbox 360 und die Playstation 3 eine doppelte Anzahl an Frames zu berechnen, wie es die obige Methode beschreibt. Es musste also eine Berechnung eingesetzt werden, die annähernd so schnell ist wie unter 2D. Hier hört es auch schon auf mit den Gemeinsamkeiten zu einem CGI-Workflow, bei dem Berechnungen in Echtzeit kein Thema sind. Wobei man erwähnen muss, dass auch Produktionen im Filmbereich von Echtzeitberechnungen profitieren können, sofern die Qualität stimmt. Spiele hingegen müssen kategorisch ein Realtime-Rendering einsetzen. Dazu wurden schon immer eine Menge Tricks während der Entwickklung angewendet, um Einsparungen geschickt zu nutzen, die faktisch keinen Unterschied machen.

Um sich nun also zwei Bilder zum Preis von einem zu erkaufen, nutzt die CryEngine 3 die Tatsache, dass man für ein stereoskopisches Bild auch mit einem Bild auskommen kann, wenn zusätzlich die Tiefeninformation der Objekte oder der einzelnen Pixel bekannt ist. Dazu wird nach der Berechnung des Bildes jedes Pixel entsprechend seiner Tiefe für die beiden finalen Bilder nach links bzw. nach rechts verschoben. Das Ergebnis sind zwei stereoskopische Bilder, die einen überzeugenden 3D-Effekt erschaffen – für den Preis von gerade mal 2% Performance-Einbuße (und ein bisschen mehr Speicher).

Leider gibt es auch einen Wermutstropfen: Bei Objekten, die sehr nah an der Kamera sind, fehlt die korrekte Parallaxe, weil die beiden Kameras nicht wirklich aus zwei unterschiedlichen Blickwinkeln aufnehmen. Weil dieser Effekt aber hauptsächlich im vom Crysis 2 nicht genutzten negativen Parallaxenbereich sichtbar – und in der Tiefe faktisch nicht mehr wahrnehmbar – ist, wird diese Einschränkung nicht erkannt.

Das Übertragungsverfahren bei Crysis 2

Der letzte Schritt für das Bild ist der Weg über einen Standard hin zum Monitor oder TV. Da es für die Playstation 3 bereits den »HDMI 1.4a«-Standard gibt, liegt es nahe, ein passendes Format zu unterstützen. Crysis 2 nutzt hier einen »S3D 720p«-Modus«. Dafür wird ein Stereobild aus einem 1280x720 Pixel großen linken Bild, einer 30 Pixel hohen Lücke und abschließend vom rechten Bild (ebenfalls 1280x720) erstellt. Dieses Paket wird als ein Frame über HDMI an den Fernseher übermittelt, der es dann entsprechend dekodiert und darstellt.

Für die Xbox 360 gibt es leider noch keinen internen Standard, sodass hier auf einen sogenannten »frame-compatible stereo mode« zurückgegriffen werden muss. Das zu übertragende Bild ist damit kompatibel zu herkömmlichen Bildstandards (720p, 1080p etc.) und wird von den involvierten Geräten nicht als Stereobild erkannt. Erst die Umschaltung am Fernseher auf den 3D-Modus macht aus dem Bild ein 3D-Bild. Für Crysis 2 unterstützen wir unter anderem Verfahren wie Side-by-Side oder auch Interlaced, die von fast allen gängigen 3D-TVs der neuen Generation verstanden werden. Zu beachten ist hier, dass bei Geräten, die nativ das Stereobild im »vertically interlaced«-Verfahren darstellen, die vertikale Auflösung der Frames auch der nativen Auflösung des Fernsehers exakt entsprechen muss. Ansonsten wird das Bild im Fernsehgerät auf die volle Größe gestaucht oder gestreckt, was den Zeilenwechsel und damit das Stereobild zerstört. Beim »Checkerboard«-Verfahren muss darüber hinaus zwangsläufig die vertikale mit der horizontalen Auflösung übereinstimmen.

Auf dem PC existiert zurzeit eine dominierende Lösung für S3D von Nvidia, die mit aktiver Shutterbrille und einem 120-Hz-Monitor stereoskopes 3D erzeugt. Die CryEngine nutzt hier zwar die Hardware, baut aber auf ihr eigenes 3D-Rendering auf. Andere Systeme werden ebenfalls unterstützt – aber für eine finale Liste ist es derzeit noch zu früh.

Zurück zum Start – obwohl Crysis 2 den Tiefeneindruck hauptsächlich hinter der Projektionsebene nutzt, muss unsere Engine auch den Effekt mit negativer Parallaxe beherrschen. Dieser Effekt kann unter kontrollierten Umständen sehr gut auch in Cutscenes eingesetzt werden. Für qualitativ hochwertige Ergebnisse hilft also die Umschaltung auf das übliche Render-Verfahren, um anschließend zwei komplette Bilder korrekt zu berechnen.

Wen betrifft S3D?

Welche Entwicklungsabteilungen müssen sich mit den neuen Herausforderungen, ein Spiel in S3D zu produzieren, in besonderem Maße auseinandersetzen? Wie oben erwähnt, sind dies zu allererst die Programmierer, sowohl auf Seiten der Engine als auch im Game Code. Vor allem das GUI kann ohne Anpassung nicht einfach übernommen werden.

Ein Spiel einfach zu »dreidimensionalisieren« reicht bei weitem nicht aus, wenn man ein echtes S3D-Game entwickeln will. Hier sind auch die Designer gefragt, die neuen Elemente sinnvoll einfließen zu lassen. Gerade an diesem Punkt sehe ich noch längere Lern- und Entwicklungsphasen, bis Spiele die neue Bildsprache umfassend einsetzen und sich auch diverse Konventionen etablieren, die Spieler verstehen und zu nutzen wissen.

Wer kauft S3D?

Ein weiterer Aspekt für S3D-Spiele ist das Bewerben dieser speziellen Eigenschaft. Da die Verbreitung von 3D-Fernsehern oder Monitoren noch gering ist, kann dem Kunden der Effekt auch nur sehr vereinzelt nahegebracht werden. Aber wie bewirbt man einen Vorteil, den man gesehen haben muss, um ihn zu verstehen? Selbst im Print-Bereich gibt es noch keine zufriedenstellende Lösung, 3D zu bewerben. Der Klassiker »Hai springt aus dem Fernsehrahmen hervor« ist eine nur behelfsmäßige Lösung, die eher belustigt als wirklich überzeugt. Anaglyphe Bilder lassen sich zwar gut abdrucken, benötigen aber eine Brille und sind weit von der tatsächlichen Wirkung des Produkts entfernt. Bis sich 3D in den Mainstream-Fernsehkanälen durchsetzt, bleibt das 3D-Kino ein gelungener Rahmen, auch andere 3D-Produkte zu bewerben. Hier ist aber ein weiterer Umstand zu bemerken: Der 3D-Tiefeneffekt verändert sich in seiner Stärke mit zunehmender Bildgröße – ein Trailer, der auf einem Monitor einen intensiven 3D-Effekt zeigt, kann auf der großen Leinwand schon zu viel sein, oder umgekehrt. Auf der anderen Seite stehen unsere Konsumenten: So wie es Menschen gibt, die farbenblind sind, verhält es sich leider auch mit dem stereoskopischen Sehen. Nicht jeder Gamer sieht den Effekt gleich gut bzw. einige sehen ihn überhaupt nicht. Manchen Spielern wird bei S3D auch übel, was auf den gleichen Effekt wie bei der Seekrankheit zurückzuführen ist. Dieser Anteil der Bevölkerung ist nicht groß, aber man sollte beachten, dass ein Spiel auch ohne S3D spielbar bleibt. Auf reine S3D-Titel werden wir wohl noch eine Weile warten müssen.

Ausblick in die S3D-Spielezukunft

Neue Technik hin oder her – sie allein wird nur wenige überzeugen, ein Spiel zu kaufen. Es ist also wesentlich mehr nötig, als einfach nur eine Dimension mehr hineinzupacken. Es ist aber auch mehr möglich, denn wie schon angedeutet, wird S3D in vielen etablierten Genres neue Impulse setzen. Für First-Person-Shooter ist es zum Beispiel die Immersion (eines dieser beliebten »Buzzwords«) – der Spieler rückt noch dichter an das Geschehen heran, kann noch intensiver in die Handlung einbezogen werden. Der Zugewinn an Information ist in der dritten Dimension nicht nur für Actiontitel hilfreich, auch für ein Strategiespiel ist S3D eine Bereicherung. Oder auch bei Sportspielen, denn wer ein reales Fußballmatch in S3D gesehen hat, ahnt wahrscheinlich schon, wie viel mehr auch ein Sportspiel auf der Konsole oder dem PC Spaß machen kann. Noch mehr Potenzial sehe ich in Genres, die vielleicht bisher wenig mit 3D zu tun hatten, wie zum Beispiel Puzzle Games. Diese werden die neue Ebene sicherlich nutzen, vielleicht sogar neue Subgenres bilden (Stereopuzzles?). Zusammenfassend werden alle Spiele in mehr oder weniger großem Umfang von der neuen Bildsprache Gebrauch machen, da es einfach Spaß bringt, mit ihr zu spielen. Nicht zuletzt auch, weil sie einfach intuitiver ist – denn unsere reale Welt ist nun mal 3D.

Die Technik, die bisher noch eine große Verbreitung verhindert hat, wird sich rasant weiterentwickeln. Auf lange Sicht werden sich autostereoskopische Displays durchsetzen, vor allem im TV-Bereich. Den Anfang werden Handhelds und Smartphones machen, die bereits nächstes Jahr Einzug halten. Eine Plattform wie der Nintendo 3DS ermöglicht es, ein Spiel auch kompromisslos auf S3D auszurichten. Und wer schon einmal über eine Kombination von S3D mit den neuen Controllern wie Kinect oder Sony Move nachgedacht hat, wird erkennen, dass hier etwas ganz Neues entstehen kann.

Entwicklern, die sich diesem neuen Feld widmen möchten, bietet sich ein vollkommen neuer Bereich, S3D als Design-Element einzusetzen. Hier ist noch echte Pionierarbeit möglich – wir können Klassiker schaffen, die ein ganzes Genre neu definieren. Ich freue mich drauf!

Alexander Taube

3D in your Face!

So, endlich lüfte ich das Geheimnis, warum wir in Crysis 2 tatsächlich keine negative Parallaxe verwenden. Dabei ist der Effekt verantwortlich für die »coolen« Momente in den 3D-Kinofilmen, wenn uns Objekte wie zum Beispiel ein Monster oder ein Projektil gefährlich nahe kommen. Dazu muss erklärt werden, dass Objekte, die aus dem Monitor scheinbar heraustreten, eine sogenannte »window violation« provozieren können. Mit dem Fenster ist hier der Rahmen der Abbildungsfläche, also des Monitors, gemeint. Wenn sich nun ein Objekt aus dem Monitor heraus und zur Seite auf den Rand des Monitors zu bewegt, wird es durch den Rahmen abgeschnitten, obwohl es optisch davor erscheint. Dieser Fehler ist für den Betrachter extrem störend. Weil in einem interaktiven Spiel die Kamerakontrolle beim Spieler liegt, ist es nicht möglich, diesen Effekt zu vermeiden, solange Objekte aus dem Monitor heraustreten.

Ein weiterer Punkt ist der oben schon erwähnte Wechsel des Fokus in der Tiefe. Liegen Objekte sehr weit auseinander, die der Spieler in schneller Folge fokussiert, ermüden die Augen rascher und mindern somit den Spaß am Spiel.

Kinect aus Entwicklersicht

Im Gespräch mit Rare und Ubisoft Montreal

Wie aufwendig ist die Entwicklung eines Kinect-Titels wirklich? Was können Hardware und Devkit? Wir fragen beim technischen Leiter von »Kinect Sports« und dem Game Designer von »Your Shape: Fitness Evolved« nach.

Nick Burton

ist Development Director bei Rare und war technischer Leiter des Kinect-R&D-Teams.

WWW
makinggames.de
- Die Bewegungssteuerung der Wii

Microsoft rechnet bis zum Jahresende mit fünf Millionen verkauften Kinect-Einheiten, Sony meldet 1,5 Millionen Move-Spieler allein in Europa. Egal wie glaubwürdig diese Zahlen nun sind: Fest steht, dass sowohl Kinect als auch Move zu den großen Gewinnern des diesjährigen Weihnachtsgeschäfts gehören und damit spätestens jetzt zu den attraktiven Plattformen gehören. Aber wie sehr muss man sich als Entwicklerstudio überhaupt umstellen und anpassen, wenn man das Thema »Motion Controlled Gaming« angehen möchte? Unsere Interviews mit Nick Burton (Rare, Kinect Sports) sowie Nicola Godin (Ubisoft Montreal, Your Shape: Fitness Evolved) liefern Ihnen hoffentlich ein paar brauchbare Antworten auf diese Frage.

Kinect aus Techniker-Sicht: Im Gespräch mit Nick Burton

Making Games Was sind die wichtigsten Dinge, über die Entwickler nachdenken sollten, bevor sie sich für ein Kinect-Projekt entscheiden?

Nick Burton Am wichtigsten ist wohl die Frage, wie flexibel man arbeiten kann. Und das betrifft nicht nur das Team, sondern auch die Engine und den Code. Es ist wenig erfolgversprechend, gleich zu Projektbeginn über Strategie, Engine oder gar ein technisches Verfahren wie PCA (Anm. der Red.: Principal Component Analysis) zu entscheiden und dann dogmatisch daran festzuhalten. Jeder spricht gerade über »Motion Controlled Gaming« und Gestenerkennung, aber nur den wenigsten ist klar, dass man zumindest bei Kinect über weitaus mehr nachdenken muss. Klar interagiert der Spieler bei »Kinect Sports« in erster Linie über Gesten. Die große Herausforderung ist jedoch, die gesamten Körperbewegungen des Spielers jederzeit korrekt zu interpretieren. Ein Beispiel: Wenn du zwei Personen sagst, sie sollen rennen, sind das zwei völlig unterschiedliche Bewegungsabläufe – schlicht und einfach weil jeder Mensch andere

Der **Speerwurf** in Kinect Sports, das Rare pünktlich zum Launch von Kinect fertiggestellt hat: »Wenn du zwei Personen sagst, sie sollen rennen, sind das zwei völlig unterschiedliche Bewegungsabläufe.«

körperliche Voraussetzungen hat. Bei Aktivitäten wie Bowling oder Tischtennis wird das Ganze natürlich noch ungleich komplexer. Also brauchst du Systeme, die entsprechend tolerant und flexibel sind. Das System muss sich dem Spieler anpassen und nicht umgekehrt, wie es bei den klassischen Steuerungsmethoden per Gamepad oder Maus und Tastatur der Fall ist. Dieser Ansatz erfordert von allen Beteiligten ein grundlegendes Umdenken.

Making Games Also mehr auf Iteration konzentrieren als aufs reine Implementieren?
Nick Burton Ja, und zwar von Anfang an. Natürlich muss man auch über die Implementierung in die Game Engine nachdenken. Man sollte diese Fragen jedoch erst mal soweit wie möglich ausklammern. Entwickle einen Prototypen, probiere ihn aus, optimiere ihn, teste ihn erneut, optimiere wieder und so weiter. Es ist extrem wichtig, alles so genau wie möglich zu analysieren: Welche Bewegungen funktionieren, wie werden sie vom System erkannt, welche Daten bekomme ich wieder heraus?

Making Games Gehen wir vielleicht noch mal einen Schritt zurück: Was muss ich denn als Entwickler überhaupt tun, um an ein Kinect-Devkit heranzukommen?
Nick Burton Das ist leicht: Du musst einfach nur deinen Account Manager kontaktieren.

Making Games Hilft es beim Kontaktieren, wenn ich schon Vorerfahrungen mit Bewegungserkennung vorweisen kann, also zum Beispiel ein Wii-Projekt im Portfolio habe?
Nick Burton Es gibt keine Voraussetzungen. Jeder, der zertifizierter Xbox-Entwickler ist, hat auch automatisch Zugriff aufs Kinect-Devkit – es ist Teil des SDK. Im Prinzip musst du also nur ein Kinect-System im Laden kaufen, es an dein Xbox-Devkit anstöpseln und kannst sofort loslegen.

Making Games Wie kompliziert ist es generell, mit dem Kinect-Devkit zu arbeiten?
Nick Burton Tatsächlich ist es sogar ziemlich einfach. Im Prinzip versorgt dich das Teil wirklich mit allen relevanten Daten, angefangen beim Image Feed über den Depth Feed bis hin zur Skeletal Data, falls du Letzteres aktiviert hast. Im Grunde ist alles so im SDK integriert, wie man es von einem normalen Controller, einer Webcam oder einem Audio Stream gewohnt ist – mit dem einzigen wichtigen Unterschied, dass all diese Dinge jetzt zentrale Bestandteile des User-Interface-Bereiches sind.

Making Games Du erwähntest, wie wichtig Prototyping bei Kinect ist. Was sind hier die Hauptunterschiede zu »normalen« Spielen?
Nick Burton Wenn du ein Gamepad als Eingabegerät verwendest, ist ein Großteil vom Interface bereits vordefiniert. Du drückst den Analogstick nach vorn, also bewegt sich auch deine Spielfigur in diese Richtung – das war's. Wenn du dagegen bei Kinect definierst, dass du die Spielfigur durch einfaches Gehen bewegst, bist du noch lange nicht fertig, sondern gerade erst am Anfang. Meine Tochter ist sechs Jahre alt, naturgemäß deutlich kleiner als ich und bewegt sich auch anders. Das hat jede Menge Implikationen, nicht nur auf technische Dinge wie die Kameraperspektive, sondern auch aufs Game Design. Dafür musst du Lösungen finden, die sich natürlich anfühlen. Und das schon in der Prototyping-Phase, damit du es auch testen kannst.

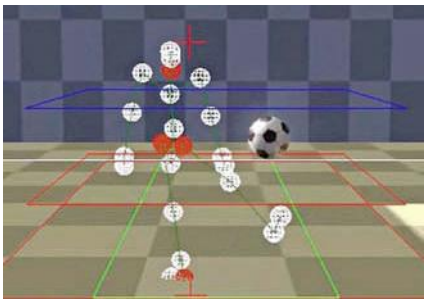
Unser Design Team verwendet dafür den Begriff »Affordance« (Anm. d. Red.: Affordance = Aufforderungscharakter), was frei übersetzt bedeutet: Das System tut exakt das, was offensichtlich ist. Erst wenn du jemanden beim realen Bowling genauestens studierst, kannst du auch wirklich die für das Spiel relevanten Bewegungsabläufe herausfiltern.

Beim Spielen mit Kinect ist eigentlich immer offensichtlich, was gerade zu tun ist. Und genau aus diesem Grund benötigt das Spiel eine sehr sorgfältig durchdachte Bewegungserkennung – übrigens nicht zu verwechseln mit einer fehlertoleranten Bewegungserkennung. Denn trotz aller unterschiedlichen Bowling-Stile muss dein System in der Lage sein, beim Werfen Daten wie Beschleunigung, Richtung und Rotation aus der skelettalen Kameraaufnahme zu extrahieren. Und zwar alle Daten gleichzeitig mit geringstmöglicher Verzögerung.

Making Games Angesichts dieser Komplexität: Wie viel sollten Entwickler tatsächlich über Physik und Kinetik wissen, wenn sie sich mit Kinect beschäftigen wollen?
Nick Burton Mit einer guten Beobachtungsgabe und einem feinen Gespür für deinen eigenen Köper und dessen Muskeln erfüllst du eigentlich schon 90% der Voraussetzungen. Ein wenig Grundwissen über Beschleunigung und Biomechanik hilft natürlich auch.
Außerdem wichtig: dein Spiel von Leuten testen zu lassen, die zuvor noch nie mit »Motion Controlled Gaming« in Berührung gekommen sind. Beobachten, was sie tun. Analysieren, was ihre Erwartungshaltung war.

Making Games Gibt es Middleware, die euch dabei geholfen hat, oder musstet ihr bei der Entwicklung von Kinect Sports quasi wieder bei null anfangen?
Nick Burton Klar verwenden wir wie jeder Entwickler Middleware. Aber auch das Kinect-SDK selbst enthält »out of the box« schon unglaublich viele Tools, Funktionen und Informationen, die dir sinnvoll unter die Arme greifen. Etwa ein Tool, mit dem du sehr detailliert die aufgenommenen Bewegungen analysieren kannst. Das SDK liefert sogar so viel Input, dass man sehr genau darüber nachdenken sollte, was man zu welchem Zeitpunkt überhaupt benötigt. Glücklicherweise kannst du alle Funktionen auch separat ansteuern, also etwa nur den »Skeletal Feed« oder die »Spracherkennung«. Es gibt aller-

Bei diesem **Sports-Prototypen** hieß Kinect noch Natal: »Alles ist so im SDK integriert, wie man es von einem normalen Controller, einer Webcam oder einem Audio Stream gewohnt ist – nur dass diese Dinge jetzt zentrale Bestandteile des User-Interface-Bereiches sind.«



Selbst dieser frühe **R&D-Prototyp** wurde ausführlich getestet: »Es gab ein Wireframe-Skelett und einen Wireframe-Ball auf einer grünen Fläche, das war's.«

dings noch keine kaufbare »All in One«-Middleware-Lösung für Kinect-Spiele, falls du das meinst.

Making Games Wie viele eurer Prototypen haben es denn ins finale Spiel geschafft und wie viele musstet ihr wegwerfen?
Nick Burton Na ja, es wurde schon sehr viel experimentiert. Wir haben sehr kleine Teams zusammengestellt – meist nur jeweils ein Programmierer, ein Artist und ein Designer – und ihnen gesagt: »Macht etwas Cooles für Kinect, ihr habt genau eine Woche.« Da kamen die verrücktesten Sachen heraus, etwa ein Kletter-Simulator oder eine »Marble Madness«-Variante. Es gab keine Vorgabe, aber was vor zwei Jahren von Anfang an herausragte, waren die kleinen Sportspiele. Das Erste, was wirklich die komplette Firma begeisterte, war Weihnachten 2008 ein Torschuss-Minigame. Dabei gab es damals noch gar keine echte Gestenerkennung, wir haben lediglich die Xbox-Kamera und eine Physik-Middleware mit Strichgrafik kombiniert. Aber es funktionierte und hat ausnahmslos jedem Spaß gemacht, wahrscheinlich weil man so wenig abstrahieren muss und einfach instinktiv weiß, was zu tun ist. Und das war dann auch der Punkt, an dem wir uns entschieden haben, in die Sportrichtung zu gehen.

Making Games Angesichts dieser Erfahrungen: Hältst du es für eine schlaue Idee, ein Kinect-Spiel für Core Gamer zu entwickeln?
Nick Burton Ich wäre zumindest vorsichtig, ganz einfach, weil bereits unsere Minigames für Kinect Sports extrem komplex in der Umsetzung waren. Das würde sich bei einem Strategiespiel oder Shooter nochmals potenzieren. Wir haben natürlich schon selbst darüber nachgedacht und bereits viele gute Ideen in der Pipeline, über die ich aber noch nicht reden kann. Aber es wäre einfach zu früh gewesen. Kinect ist eine komplett neue Technologie, an die man sich als Entwickler und auch als Spieler naturgemäß erst gewöhnen muss. Ich glaube zum Beispiel, dass das iPhone nie ein solcher Erfolg geworden wäre, wenn es das Prinzip des Touchscreens nicht schon vorher gegeben hätte.

Making Games Wie habt ihr die ganzen Iterationen überhaupt organisiert? Gab es da Unterschiede im Workflow verglichen mit euren früheren Projekten?
Nick Burton Ja, die gab es. Wir haben dafür eigens ein neues Workflow-System namens RITE eingeführt. RITE steht dabei für Rapid, Iteration, Test und Evaluate. Das bedeutet vor allem, dass wir bereits mit extrem frühen Pototypen in die User Research eingestiegen sind. Es gab ein Wireframe-Skelett und einen Wireframe-Ball auf einer grünen Vektorfläche, das war's. Erst als wir die zentralen Interaktionsmechanismen genau dort hatten, wo wir sie haben wollten, wurden weitere Dinge hinzugefügt. Das Testen und die Evaluierung liefen während der gesamten Projektphase im Wochen-Rhythmus, und zwar jeweils mit Kindern, Erwachsenen sowie Senioren. Gegen Projektende haben wir die Playtesting-Frequenz sogar noch mal erhöht.

Making Games Wie habt ihr das denn räumlich hinbekommen?
Nick Burton Das war tatsächlich ein Problem. Bei normalen Spielen kannst du ja ziemlich viele Menschen in einen ziemlich kleinen Raum stopfen. Angesichts des benötigten Bewegungsradius bei Kinect wird das naturgemäß nicht funktionieren. Außerdem musst du in Betracht ziehen, dass Spielen mit Kinect wesentlich schneller ermüdet als mit einem Gamepad. Deswegen haben wir zum Beispiel Schichten eingeführt und Paare alternierend testen lassen. Während der eine mit vollem Körpereinsatz spielt, dokumentiert der andere seine Testergebnisse.

Making Games Was wären deine drei persönlichen Ratschläge für Entwickler, die sich mit Kinect beschäftigen wollen?
Nick Burton Der erste ist auf jeden Fall »Rapid, Dirty Prototypes«! Versuche nicht, etwas besonders Hübsches zu entwerfen – Hauptsache, es funktioniert. Und ich rede hier wirklich von Wireframe-Optik. Erst wenn die Kernaktionen passen, kannst du über andere Dinge nachdenken. Idealerweise kapselst du das Protyping-Team sogar ab vom Rest, damit sie nicht von Funkelzeug wie Partikeleffekten abgelenkt werden. Zweitens: Sei ehrlich zu dir selbst und zu anderen, wenn etwas nicht so funktioniert, wie es funktionieren sollte. Und habe den Mut, auch mal etwas zu verwerfen. Und der dritte Tipp wäre, regelmäßig ein paar Leute zu versammeln, über das gleiche Problem nachdenken zu lassen und zu schauen, ob sie auch auf die gleiche Lösung kommen. Falls nicht, siehe zweitens.

Nicola Godin

ist Game Designer bei Ubisoft Montreal.

Kinect aus Designer-Sicht: Im Gespräch mit Nicola Godin

Making Games Wann habt ihr angefangen, euch bei Ubisoft Montreal mit »Motion Controlled Gaming« zu beschäftigen?
Nicola Godin Alles begann 2008, als uns eine israelische Firma namens PrimeSense eine 3D-Kamera präsentierte. Ubisoft war sehr interessiert an dieser Technologie und erkannte das Potenzial für Spiele. Von da an haben wir sehr viel Research & Development in diese Richtung unternommen. Der Plan war ursprünglich, ein Spiel zusammen mit der Kamera auszuliefern. Aber dann kaufte Microsoft die Technologie und entwickelte daraus Project Natal bzw. Kinect. Und das hat natürlich einiges verändert. (lacht)

Making Games Und wann habt ihr mit dem Projekt »Your Shape: Fitness Evolved« begonnen?

Nicola Godin Das war ebenfalls 2008. Die ersten Prototypen gingen eigentlich eher in Richtung Tanzspiel. Weil wir aber Kamera und Spiel ursprünglich als Bundle verkaufen wollten und somit natürlich etwas teurer gewesen wären, mussten wir etwas finden, was dem Spieler einen echten Mehrwert liefert. Und so kamen wir auf die Idee, ein Fitnessspiel zu entwickeln – schließlich geben die Leute Hunderte von Dollar im Jahr für Fitnessstudios aus.

Making Games Wie wertvoll war es für euch, durch diese ganze Vorgeschichte schon sehr früh mit dem Konzept der Ganzkörper-Erkennung in Berührung gekommen zu sein?
Nicola Godin Extrem wertvoll. Je mehr Zeit man als Entwickler hat, eine neue Technologie kennenzulernen, desto besser. Die ursprüngliche 3D-Kamera erkannte nur Silhouetten, Kinect ergänzte das später ja noch mit »Skeletal Tracking«. Da wir Ersteres schon kannten, fiel es uns natürlich erheblich leichter, beide Konzepte sinnvoll in einem Spiel zu kombinieren.

Making Games Hattet ihr also schon ein eigenes Toolset, bevor es überhaupt die ersten Kinect-Devkits gab?
Nicola Godin Ja, wir haben sogar in Zusammenarbeit mit PrimeSense eigene Tools für die 3D-Kamera und eine komplett neue Engine entwickelt.

Making Games Wie schwierig war es dann für euch, alles wieder an Kinect anzupassen?
Nicola Godin Wir hätten das Spiel ohnehin auf der Xbox 360 veröffentlicht, insofern waren wir gut vorbereitet. Es gab also keine größeren Schwierigkeiten, die grundlegende Technologie ist ja gleich geblieben.

Making Games Würdest du nach euren Erfahrungen den Entwicklern raten, grundsätzlich ein eigenes Toolset zu entwickeln? Oder reicht das Kinect-Devkit vollkommen aus?
Nicola Godin Das hängt in erster Linie davon ab, was man überhaupt machen möchte. Für viele Dinge ist das Kinect-Devkit vollkommen ausreichend, aber häufig hat man als Designer ja noch ein paar eigene Ideen, die über die Grundfunktionen hinausgehen. Gerade mit Kinect kann man wirklich die verrücktesten Dinge anstellen. Und je verrückter die Idee, desto notwendiger werden auch die eigenen Tools.

Making Games Was sind für dich die größten Unterschiede verglichen mit dem Design von traditionellen Spielen?
Nicola Godin Es gibt keinen Controller mehr, insofern musst du wirklich alles komplett neu erfinden. Allein für die Menüs haben wir insgesamt sechs Prototypen benötigt, um die für uns perfekte Lösung zu bekommen. Auch die unterschiedlichen Körpermaße der Spieler sind eine besondere Herausforderung, weil sie nicht nur das Tracking, sondern auch sämtliche Menüs und Gameplay-Elemente beeinflussen.

Das Fitnessspiel **Your Shape:** »Da wir Kamera und Spiel ursprünglich als Bundle verkaufen wollten und somit etwas teurer gewesen wären, mussten wir etwas finden, was dem Spieler einen echten Mehrwert liefert.«

Your Shape verfügt über ein eigenes **Kalibrierungstool:** »Die unterschiedlichen Körpermaße der Spieler sind eine besondere Herausforderung, weil sie nicht nur das Tracking, sondern auch sämtliche Menüs und Gameplay-Elemente beeinflussen.«

Making Games Welches Know-how braucht man dazu als Game Designer?
Nicola Godin Als Game Designer eines Spiels mit Bewegungserkennung ist es vor allem wichtig, den eigenen Körper zu kennen. Du musst wissen, wie er funktioniert, wie du mit ihm arbeiten kannst. Ich habe mich zum Beispiel viel mit zeitgenössischen Tänzen beschäftigt, spiele regelmäßig Badminton und war Musiker. All diese Hobbys haben mir sehr geholfen, ein flüssiges Spielgefühl sicherzustellen – und das ist bei der Bewegungserkennung ein ganz entscheidender Spaßfaktor. Ein gutes Körpergefühl kann man lernen! Wir haben natürlich auch mit vielen Fitness- und Kinetik-Experten zusammengearbeitet, um ein effizientes Trainingsprogramm zusammenzustellen. Aber das ist ja nichts Besonderes: Wenn du einen Militär-Shooter entwickelst, sprichst du ja auch mit Militär-Experten.

Making Games Und wie schaut's mit dem Rest des Teams aus? Gab's spezielle Fortbildungen?
Nicola Godin Ja und nein. Die Arbeit für die Pro-

»Das Wichtigste beim **Block-Minigame** war, dass der Spieler auch wirklich ein körperliches Gefühl dafür entwickelt, wann er die Blöcke trifft und wann nicht.«

grammierer ist größtenteils die gleiche wie bei traditionellen Spielen – Code bleibt nun mal Code. Aber auch sie müssen ein Verständnis dafür bekommen, wie der Körper funktioniert. Dafür reicht es aber aus, wenn sie das Ergebnis ihres Codes so häufig wie möglich selbst ausprobieren. Ein spezielles Training braucht es dafür eigentlich nicht.

Making Games Wie viel bei Your Shape war erprobtes Konzept, wie viel Experimentieren und wie viel reines Glück?
Nicola Godin Glück passiert zwar manchmal, aber leider viel zu selten. Wenn du ein Spiel für eine neue Technologie entwickelst, ist es grundsätzlich eine gute Idee, mit dem Einfachstmöglichen zu beginnen. Bei uns war das zum Beispiel ein Minigame, in dem du Blöcke zerschlagen musst. Dem haben wir dann nach und nach neue Gameplay-Elemente hinzugefügt und geschaut, was gut funktioniert und was weniger. Ein simpler Prozess von Ja oder Nein mit immer wieder neuen Versuchen. Das Wichtigste beim Block-Minigame war etwa, dass der Spieler auch wirklich ein körperliches Gefühl dafür entwickelt, wann er die Blöcke trifft und wann nicht. Glück kommt vor, und manchmal führt sogar ein Bug oder Fehler zu einer genialen Idee. Ich würde mich aber lieber auf das Ausprobieren und Spielen verlassen. (lacht)

Making Games Your Shape: Fitness Evolved richtet sich wie alle Kinect-Starttitel vor allem an Casual Gamer. Werden wir demnächst auch Spiele für Hardcore Gamer sehen?
Nicola Godin Im Moment konzentriert sich Microsoft tatsächlich sehr bewusst auf Casual Gamer. Ich persönlich sehe Kinect aber fast als komplett neue Konsole. Man kann so unglaublich viele coole Dinge damit anstellen. Also ja, Spiele für Hardcore Gamer werden definitiv kommen. Aber auch sie werden deutlich anders sein als das, was wir aktuell spielen. Wie bewegst du zum Beispiel einen Charakter durch ein Kinect-Rollenspiel? Willst du den Spieler vor der Konsole wirklich auf der Stelle laufen lassen? Nein, du solltest dir definitiv etwas anderes einfallen lassen. Alle Genres müssen also zwangsläufig neu erfunden werden. Da werden definitiv einige coole Konzepte und Überraschungen auf uns zukommen. Die ersten Spiele für Hardcore Gamer werden aber wahrscheinlich ein Hybriden aus Controller- und Bewegungssteuerung sein. Also zum Beispiel Shooter, in denen du ganz normal mit dem Gamepad zielst, die Granaten aber wie in der Realität mit der Hand wirfst.

Making Games Was würdest du Entwicklern raten, die morgen mit einem Kinect-Projekt anfangen wollen?
Nicola Godin Der erste Ratschlag ist simpel, aber ganz entscheidend: Kennt euren eigenen Körper! Der zweite wäre: Plant erheblich mehr Zeit ein, als ihr glaubt, was ihr brauchen werdet! Es wird garantiert jede Menge Überraschungen geben, wenn ihr mit Kinect arbeitet. Drittens: Holt euch so viel Input von außen wie möglich und investiert massiv ins Playtesting. Denn gerade beim ersten Kinect-Projekt kann es sehr leicht passieren, dass man im Developer-Tunnel landet und den Fokus verliert. Als Entwickler weißt du schließlich, welche Eingabeinformation die 3D-Kamera sehen will, und entsprechend verhältst du dich auch beim Testen. Viel entscheidender ist es jedoch herauszufinden, wie jemand spielt, der nicht über diese Informationen verfügt. Das hilft dir nicht nur beim Optimieren der Bewegungserkennung, sondern liefert dir häufig sogar neue Gameplay-Ideen – etwa wenn die Tester etwas völlig anderes machen, als das Minigame vorgibt, und trotzdem oder gerade deshalb jede Menge Spaß dabei haben.

Making Games Apropos Spaß: Was für eine Art von Spiel würdest du dir persönlich für Kinect wünschen?
Nicola Godin Ein Horrorspiel wäre der Hammer! Das bietet sich auch förmlich an, weil die große Stärke von Kinect ja die Immersion des Spielers ist. Schließlich gibt's keinen Controller mehr, der Distanz und Abstraktion schafft. Und je besser die Immersion, desto besser funktioniert natürlich auch Horror. Stress- und Angstsituationen würden einfach viel intensiver wirken, wenn du direkt im Spiel bist und dich etwa vor einem wahnsinnigen Axtmörder verstecken musst. Das wäre mal spannend und interessant!

Making Games Wow, da kommt dann also noch einiges auf uns zu.
Nicola Godin Darauf kannst du wetten! Wir stehen erst am Anfang, was Bewegungserkennung im Allgemeinen und Kinect im Speziellen angeht. Unserer Kreativität sind im wahrsten Sinne des Wortes keine Grenzen gesetzt.

Interviews: Heiko Klinge

NextGen für den Browser

Die Technik von Drakensang Online

Das Action-Rollenspiel Drakensang Online ist in vielerlei Hinsicht ein Pilotprojekt: Das erste MMO unseres Studios Bigpoint Berlin, das zuvor unter dem Namen Radon Labs die großartigen Solo-Rollenspiele Drakensang sowie Drakensang: Am Fluss der Zeit geschaffen hat. Die erste Bewährungsprobe für die von Grund auf neu entwickelte Grafik-Engine Nebula 3. Und vor allem soll Drakensang Online das erste Browsergame werden, das eine ähnliche optische Qualität und Detailverliebtheit liefert wie vergleichbare Offline-Titel vom Schlage eines Titan Quest oder Sacred 2. André Weißflog, Head of Game Production, erklärt, mit welchen technischen Kniffen er und sein Team dieses Ziel erreichen wollen.

ERST PORTIEREN ...

André erinnert sich: »Die Nebula-3-Engine haben wir ursprünglich für NextGen-Konsolen und Multicore-CPUs entwickelt.« Entsprechend verfügt sie auch über sämtliche Features, die man von einer modernen Grafiktechnologie heutzutage erwartet – egal ob Multi Threading, Normal Mapping, Physik-Simulation oder weiche Echtzeitschatten. »Und wir veranstalten einen ziemlichen Voodoo mit Pixel Shadern«, so André. Besonders stolz sind er und sein Team darauf, dass sie beliebige Subsysteme in einzelne Threads auslagern können. Für das Rendering verwenden sie dabei einen Light Prepass Shader – eine Variante von Deferred Shading, die aber nicht ganz so Ressourcen-hungrig ist. Nebula 3 wurde als Highend-Egine entwickelt und bleibt auch im Browser eine Highend-Engine. Dennoch war das Team überrascht, wie schnell sich die Technologie portieren ließ. André erinnert sich: »Die klassische Engine im Browser rendern zu lassen war gerade mal eine Sache von nicht mal zwei Wochen.«

... DANN OPTIMIEREN

Trotz der kurzen Portierungszeit hatte das Team natürlich noch viel Arbeit vor sich. André erklärt: »Der Knackpunkt war die Datenverwaltung. Deshalb haben wir das Ressourcen-Streaming komplett umgebaut.« Das Ergebnis: Drakensang kommt mit lediglich drei Prozent der Daten der Offline-Vorgänger aus, ohne dass die Qualität darunter leidet. So wurden etwa 3D-Objekte in kleinstmögliche Bausteine unterteilt, die sich vielfältig kombinieren lassen. Auch die Vogelperspektive hilft beim Datensparen. Die Engine lädt nur das, was unmittelbar benötigt wird. Durch die niedrigen Datenmengen konnten auch die Turnaround-Zeiten maßgeblich verkürzt werden. So mussten die Level Designer bei den Offline-Titeln häufig mehr als eine halbe Minute warten, bis die Geometrie geladen war – bei Drakensang Online können sie nach wenigen Sekunden loslegen. »Dadurch wird nicht nur die Produktivität erhöht, die Leute haben vor allem auch viel mehr Spaß an ihrer Arbeit«, so André. Rund fünf Monate haben André und sein Team noch Zeit für weitere Optimierungen, im April 2011 wird Drakensang Online live gehen. Das technische Ziel: »Maximal 5 Megabyte Datendownload bis zum Spielstart und nur wenige Sekunden Wartezeit, bis du das erste Monster bekämpfst.« Das spielerische Ziel: »Der heilige Gral für das gesamte Team ist es, wenn die Drakensang-Fans der ersten Stunde auch das neue Spiel annehmen und ein Teil unserer Vision werden.«

DRAKENSANG-ONLINE.DE

BIGPOINT

PC-only war vorgestern
Cross-Platform-Entwicklung

Olliver Heins erklärt anhand von Bigpoints ToonRacer, wie ein Spiel von Beginn an auf unterschiedlichste Plattformen optimiert wird, welche Middleware sich am besten eignet und warum ein Comic-Rennspiel deutlich einfacher parallel zu entwickeln ist als sein fotorealistisches Pendant.

Olliver Heins

ist Producer bei Bigpoint.

Olliver hat Game Design sowie 3D Art & Animation in Berlin studiert und währenddessen schon bei Bigpoint gearbeitet. Vor seinem Studium hat er eine Ausbildung zum Mediengestalter absolviert. Seit mittlerweile zwei Jahren ist er als Producer tätig, unter anderem für das in diesem Artikel erwähnte Comic-Rennspiel ToonRacer.

Die Cross-Plattform-Entwicklung ist besonders im Bereich der Online Games ein sehr spannendes Feld. Bei Bigpoint beschäftigen wir uns schon seit einiger Zeit mit dieser Thematik, denn wir glauben nicht an die Zukunft einer einzelnen Technologie. Stattdessen wird die Vielseitigkeit für die Spieler entscheidend sein. Die Möglichkeit, in vielen unterschiedlichen Situationen und an verschiedenen Orten zu spielen, gibt nicht nur ihnen viele neue Möglichkeiten, sondern auch den Designern. Für uns als Browsergame-Entwickler stehen natürlich die mobilen Geräte besonders im Fokus, seien es die Apple-Produkte oder die Vielzahl der Android-Geräte am Markt. Das Ziel ist es dabei immer, unseren Spielern Erlebnisse über den PC hinaus zu ermöglichen. Mit »ToonRacer« bringen wir nun ein Arcade-Rennspiel heraus, bei dem wir in Sachen Cross-Platform-Entwicklung bereits gute Ergebnisse erzielen konnten.

Die technische Seite

Cross-Plattform-Entwicklung ist eine zeitaufwendige Angelegenheit, schließlich müssen gleich mehrere Systeme mit ein und derselben Software ausgestattet werden. Dabei müssen wir nicht nur die Leistungsfähigkeit der Zielplattform berücksichtigen, sondern auch all ihre Besonderheiten in Planung und Entwicklung einbeziehen.

PC, Mac, Wii, PS3, Xbox 360, IOS und Android stellen derzeit wohl die lukrativsten Plattformen für die Spiele-Entwicklung dar und sollten daher stets für das eigene Produkt in Betracht gezogen werden. Gleichzeitig haben alle diese Plattformen naturgemäß ein eigenes Betriebssystem, ein eigenes IO-System (z. B. Anzeige, Sound, Dateizugriff, Vernetzung usw.), eine unterschiedliche Byte-Order – die sogenannte Endianess (Netzwerk-Pakete etc.) – und weitere kleine gemeine Unterschiede.

Ein gängiges Verfahren für die Erkennung einzelner Plattformen ist der Einsatz von Build-Systemen wie etwa »CMake«. Dabei hat man als Entwickler die Möglichkeit, Teile des Quellcodes zu ignorieren oder fertige Bibliotheken je nach Plattform einzubinden. In gemeinsam genutzten Quelltexten verwendet man dann die Pre-Processor-Direktiven (Abbildung 1).

```
#ifdef UNIX
// Quelltext für Unix...
#endif
#ifdef WIN32
// Quelltext für WIN32...
#endif
```

Abbildung 1: Die **Pre-Processor-Direktive** im gemeinsam genutzten Quelltext.

All diese Hilfen haben wir bei ToonRacer glücklicherweise nur im Backend eingesetzt, um den Server kompatibel zu alternativen Konfigurationen zu halten – sei es beim Wechsel von Linux auf Windows oder um CPU-Berechnungen auf die GPU auszulagern.

Schön ist es da, wenn man auf High-Level-Bibliotheken oder Laufzeitumgebungen aufbauen kann, die einem diese Arbeit zum größten Teil abnehmen. Bekannte Bibliotheken wie QT, wx-Widgets oder GTK+ helfen beispielsweise dabei, GUI-Komponenten plattformübergreifend zu nutzen. Hilfreich ist auch das Mono-Runtime, das immer stärker als frischer Ersatz für Java zum Einsatz kommt.

Die Middleware-Frage

Serverseitig benutzen wir unser eigenes C++-Framework, um die Kompatibilität sicherzustellen. Zusätzlich kommt LUA zum Einsatz, was uns die besonders schnelle Implementierung der Game-Logik ermöglicht. Durch die Implementierung der Unity3D-Engine haben wir außerdem die Möglichkeit eines Cross Compilings per Mausklick. Unity3D ist in etwa mit ShiVa 3D vergleichbar, wobei die StoneTrip-Engine noch mehr Möglichkeiten für Programmierer bietet. Dafür erfreut sich Unity3D eines höheren Bekanntheitsgrades.

Unity nutzt C#, JS und Boo im Zusammenspiel mit Mono, während Shiva LUA und C++ einsetzt, ganz wie auf unserem Game Server. Wir haben beide Engines evaluiert und waren jedes Mal begeistert von der Leistungsfähigkeit gegenüber zeitaufwendigen nativen Lösungen. Während wir anfangs genau wie in C++ »defines« für die einzelnen Plattformen anlegen mussten, hat Unity mittlerweile eine komfortable Variable namens »Application.platfom« eingeführt. Diese wird entsprechend auf die folgenden Plattformen geprüft:

OSXEditor ..... In the Unity editor on Mac OS X.
OSXPlayer ..... In the player on Mac OS X.
WindowsPlayer .....In the player on Windows.
OSXWebPlayer ..... In the web player on Mac OS X.
OSXDashboardPlayer ..... In the Dashboard widget on Mac OS X.
WindowsWebPlayer In the web player on Windows.
WiiPlayer .....In the player on Nintendo Wii.
WindowsEditor .....In the Unity editor on Windows.
IPhonePlayer .....In the player on the iPhone.
XBOX360 .....In the player on the Xbox 360.
PS3 ..... In the player on the PlayStation 3
Android ..... In the player on Android devices.

Mit dieser Methode entfällt für uns ein großer Teil der Arbeit und wir können uns voll auf das Gameplay konzentrieren. Einzig Stolpersteine wie die Endianness (Byte-Order) müssen beachtet werden (Abbildung 2) – so konvertiert die Network Byte Order (Big-Endian) zu Little-Endian (PC-Systeme) und zurück.

Abbildung 3 zeigt einige der Client-Funktionen in C#, die die Bytes konvertieren. Der C#-BitConverter kann dabei angeben, welche Byte-Order das System verwendet.

Cross Platform Gaming

Ein durchaus komplexer Punkt ist es, die unterschiedlichen Plattformen im Netzwerk gegeneinander antreten zu lassen. Weil dabei die Chancengleichheit der alles entscheidende Aspekt ist, sind Unterschiede in der Auflösung und der Controller-Eingabe ernst zu nehmende Schwierigkeiten. Für die Entwickler bedeutet der Ausgleich dieser Handcaps einen extrem hohen Testaufwand. Man sollte unbedingt ausreichend Zeit für diese Testphasen einplanen, weil gerade bei den mobilen Netzwerken nicht immer optimale Bedingungen herrschen.

Weitere Anpassungen

Bei der Asset-Gestaltung verwenden wir ganz bewusst gemalte Texturen, weil diese überall gleich aussehen. Denn bei fotorealistischen Spielen müsste man etwa die Skin Shader und Umgebungseffekte so herunterskalieren, dass das Aussehen auf mobilen Plattformen erheblich leidet. In manchen Fällen ist das Ergebnis sogar so schlecht, dass auch das Gameplay nichts mehr herausreißen kann. Auch der Wechsel der Engines wird durch gemalte Texturen einfacher, weil die Assets nicht an Shader gebunden sind, sondern schon ohne großartige Beleuchtung und Effekthascherei gut aussehen.

Bei Animationen (Bones Count) und der Polygon-Anzahl lohnt es sich, eine Low-End-Version zu erstellen, die man sowohl für entfernte Objekte auf den leistungsstarken Rechnern einsetzen als auch für Plattformen mit starken Beschränkungen verwenden kann. Damit sind vor allem Nintendos Wii und das Android-IOS gemeint. Darüber hinaus kann es bei mobilen Geräten einen positiven Effekt auf die Performance

Mit **ToonRacer** bringt Bigpoint demnächst ein Comic-Rennspiel auf den Markt, das von Anfang an über mehrere Plattformen hinweg entwickelt wurde und Duelle zwischen iPad- und PC-Spielern ermöglicht wird.

geben, wenn man etwa die Kollisions- und Level-Geometie zerteilt. Während ein durchschnittlicher Gamer-PC noch eine durchaus akzeptable Performance zeigt, wenn man größere Meshes kombiniert und die Draw Call-Anzahl verringert, kann beispielsweise ein iPhone schon aufgrund der zu hohen Polygon-Anzahl (10.000 beim iPhone 3G, 25.000 beim 3GS) an Performance verlieren. Im Allgemeinen profitieren alle Plattformen von verringerten Draw Calls. Als Entwickler muss man allerdings abwägen, ob eine PVS-Map oder die Zerteilung des Levels in Segmente am Ende Arbeit spart.

Ein interessanter Punkt sind auch Komponenten, die nicht direkt mit dem Spiel zu tun haben, aber dennoch unbedingt benötigt werden. Dazu zählen zum Beispiel der Login und ein eventuell vorhandener Item-Shop. Hier muss man dann versuchen, den kleinsten gemeinsamen Nenner zu finden, um nicht alles extra für jede Plattform neu erstellen zu müssen.

Olliver Heins

```
short[] s = new short[] {0x0002, 0x0302};

//Little-Endian 02 00  02 03
//Big-Endian    00 02  03 02
```

Abbildung 2: Die Zuweisung beim Problemfall **Big-Endian und Little-Endian**.

```
public class ProtocolBuffer
        {
                public static readonly int ProtocolHeaderSize = 6;
                public static readonly Endian SystemEndian = BitConverter.IsLittleEndian ? Endian.Little : Endian.Big;
                private ProtocolHeader header;
                private Endian byteOrder = Endian.Little;

                public void WriteUShort(ushort value)
                {
                        if (this.DoEndianSwap())
                        {
                                value = ProtocolBuffer.SwapByteOrder((ushort)value);
                        }

                        this.AddByte(BitConverter.GetBytes((ushort)value));
                }

                public short ReadShort()
                {
                        if (!this.BytesAvaiable(sizeof(short)))
                                return 0;

                        short value = (short)BitConverter.ToInt16(this.bMessage, (int)this.position);

                        if (this.DoEndianSwap())
                        {
                                value = ProtocolBuffer.SwapByteOrder((short)value);
                        }

                        this.position += sizeof(short);

                        return value;
                }

                protected bool DoEndianSwap()
                {
                return (this.byteOrder != SystemEndian);
                }
}
```

Abbildung 3: Einige **Client-Funktionen** zur Konvertierung der Bytes in C#.

Aus eins mach tausend
Automatisierte Content-Generierung

Upjers-Artist Jochen Winkler erklärt, wie Sie durch automatisiertes Austauschen aus nur wenigen Assets eine gigantische Grafik-Bibliothek zaubern, und zwar ohne dass es auf Kosten der optischen Abwechslung geht.

Jochen Winkler

ist Grafiker bei Upjers.

Nach sechs Semestern Mathematikstudium wechselte Jochen Winkler auf die Games-Academy in Berlin und absolvierte die Ausbildung zum Game-Artist. Seitdem ist er als Grafiker tätig und nutzt seine Doppelqualifikation, um ungewöhnliche Probleme mit Algorithmen oder Formeln zu lösen. Jochen Winkler ist seit 2009 Teil des Upjers-Teams. Dort arbeitete er unter anderem am Social Game »My Free Farm« und übernahm bei »Kapibados« die grafische Leitung.

WWW
makinggames.de
- Cinematics aus Ingame-Assets

Die Zeiten, in denen jedes noch so kleine Team auf magische Weise Triple-A-Titel produzieren sollte, sind glücklicherweise vorbei. Und trotzdem wünscht man sich als Entwickler mit einem kleinen Team häufig größere Mengen an Content – nur wo soll der herkommen? Selbstlose Grafiker schlagen sich die Nächte um die Ohren, Kommunikationsfreudige heuern Outsourcing-Studios in Indien an, und Skrupellose verschleißen bisweilen 20 bis 30 Praktikanten für ein Projekt – so die meist schlechteren Alternativen. Glücklicherweise gibt es eine Lösung für das Problem: automatisch generierten Content.

Im Prinzip geht es darum, eine bestehende Auswahl von Elementen immer neu zu mischen. Kompliziert wird es erst dann, wenn man den Ergebnissen den »generischen« Look nicht mehr ansehen soll. In diesem Tutorial werden wir von einem einfachen Beispiel ausgehen und Schritt für Schritt den Weg hin zu einem individuelleren, generierten Content beschreiben. Die Vorgaben: Für ein Spiel werden Bilder von 2.000 unterschiedlichen Weltraum-Kolonien benötigt. Eine Kolonie besteht aus einer Funkstation, einem Lagersilo und einer Bar. Wichtig: Jedes Bild soll dabei einzigartig sein.

Wer häufiger Content generiert, verfügt in der Regel bereits über einige Skripte für Photoshop und seine 3D-Software, um sich die Arbeit zu erleichtern. Im Rahmen dieses Einstiegstutorials werden wir jedoch auf Skripte verzichten und die gesamte Mechanik zum Mischen der Einzelelemente mit herkömmlichen Animationswerkzeugen abdecken.

Modelling

Wir gestalten die Gebäude schnell aus einfachen Grundformen, auch die UV-Maps werden ignoriert. Allerdings modellieren wir von jedem Gebäude mehrere Versionen (Abbildung 1). Nur zwei Arten von Details sind sinnvoll: zum einen solche, an denen der Betrachter die Größe erahnen kann – wie etwa Türen, Fenster, Geländer oder kleine Fahrzeuge; zum anderen Elemente, an denen wir kleine Lichter anbringen können. Alle Gebäude sollten auf denselben Grundriss passen.

Als Nächstes modellieren wir Basen, auf die man die Gebäude aufstecken kann. Das »Zusammensteckprinzip« fällt weniger auf, wenn der Schnitt zwischen Gebäude und Basis schief ist. Die Basen müssen die Form einer langen Säule haben, denn später wird die Höhe des Bodens stark variiert.

Animationstools zweckentfremden

Zeitleisten, Frames und Keyframes wurden ursprünglich für Animationen erfunden, aber sie eignen sich auch perfekt zum Durchtauschen

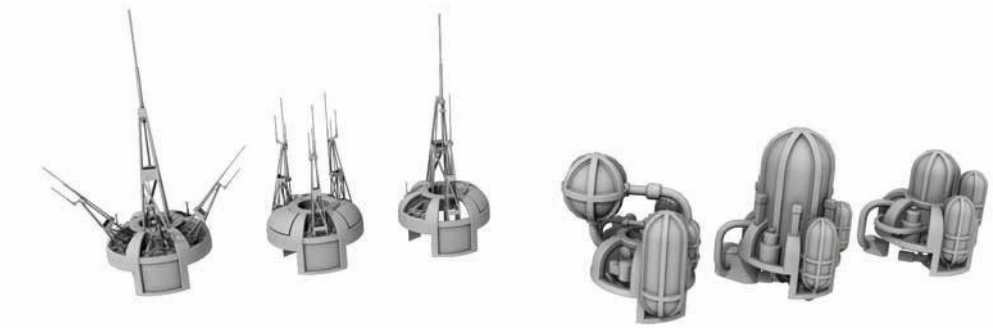
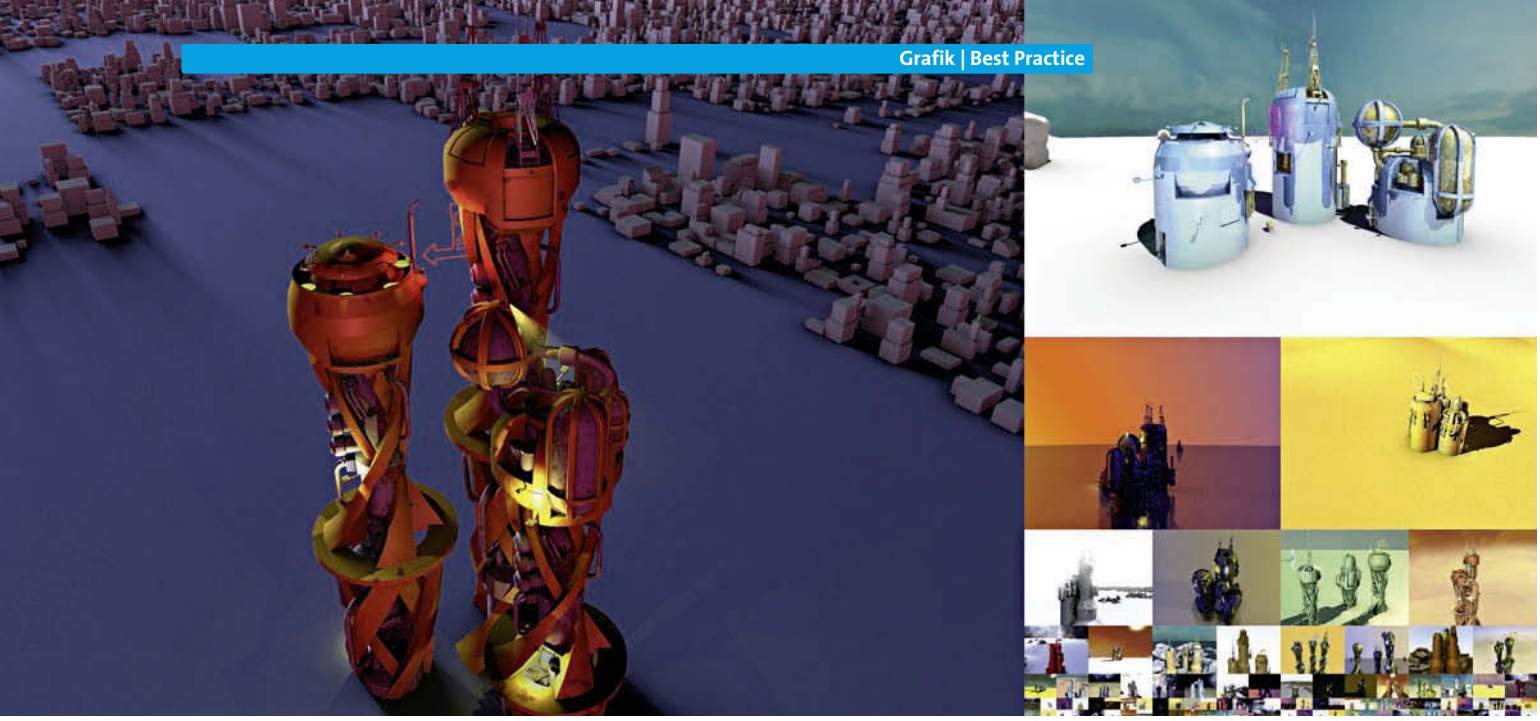


Abbildung 1: Je **drei Gebäudevarianten** vom Funkturm und der Silo-Anlage, noch ohne Lichter. Die Silhouetten sind unterschiedlich, Silos und Funktürme klar als solche erkennbar. Alle Gebäude passen auf dieselben Sockel.



Ein einzelner Grafiker kann mit vergleichsweise einfachen Mitteln **Tausende Bilder aus wenigen Elementen generieren.** Dabei spielt neben dem kreativen Einsatz von Licht und Schatten auch die intelligente Verschachtelung der Gebäudeteile eine Rolle.

von Elementen. In unserem konkreten Fall verwenden wir Keyframes, um Positionen, Sichtbarkeiten und Texturen der einzelnen Objekte zu wechseln. Für jeden Frame wird sich so ein neues Gesamtbild ergeben.

Das Durchwechseln und die Primzahlen

Beim Durchwechseln der Varianten sollte man möglichst unterschiedliche (und nicht zu kleine) Primzahlen wählen (Abbildung 2). Denn die erste Wiederholung einer Variantenkombination errechnet sich aus dem kleinsten gemeinsamen Vielfachen. Bei zwei Primzahlen ist dies einfach das Produkt, bei anderen Zahlenkombinationen kann es jedoch deutlich niedriger liegen, was natürlich nicht in unserem Sinne ist.

Ein Beispiel: Ein Turm hat 8 Modellvarianten und 12 Texturen, die immer der Reihe nach durchgewechselt werden. Es erscheinen deshalb immer die 24 gleichen Kombinationen von Beleuchtung und Textur. Wenn man aber stattdessen 7 Texturen und 11 Beleuchtungsvarianten hat, beginnt die Wiederholung erst bei 77. Es ist übrigens vollkommen in Ordnung, einzelne Elemente (beispielsweise die schönsten) doppelt zu zeigen, um auf eine neue Primzahl zu kommen.

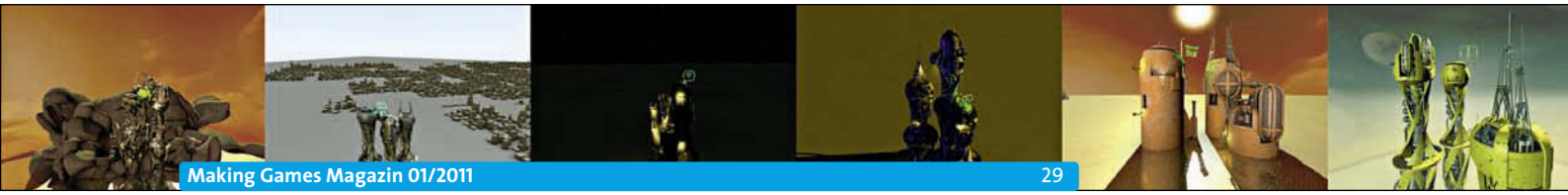
Positionierung und Sichtbarkeiten

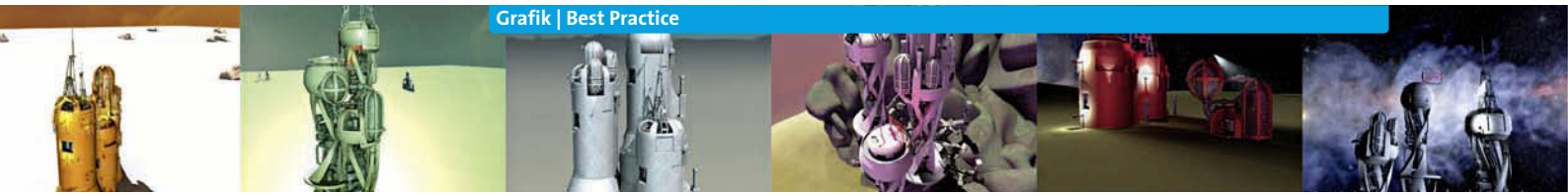
Zunächst fassen wir die Varianten der Gebäude und den Sockel in einer Gruppe zusammen. Die Gebäude werden so positioniert, dass alle an derselben Stelle im Sockel stecken. Auch die Sockel sollten sich an derselben Stelle überlagern.

Dann werden die Gruppen ihrerseits zu einer übergreifenden »Hauptgruppe_A« zusammengefasst. Aus »Hauptgruppe_A« erstellen wir zwei Kopien namens »Hauptgruppe_B« und »Hauptgruppe_C«. Die beiden neuen Hauptgruppen schieben wir etwas beiseite und kümmern uns um die Sichtbarkeit der Gebäude. Auf Frame 1 sollte in »Hauptgruppe_A« nur die Bar sichtbar sein, in »Hauptgruppe_B« das Lager und in »Hauptgruppe_C« die Funkstation. Auf Frame 2 sind die Sichtbarkeiten auf A: Bar, B: Funkstation und C: Lager. Für die anschließenden Frames ergen sich daraus die folgenden Kombinationen und Sichtbarkeiten:

Frame 1	A: Bar	B: Lager	C: Funkstation
Frame 2	A: Bar	B: Funkstation	C: Lager
Frame 3	A: Lager	B: Bar	C: Funkstation
Frame 4	A: Lager	B: Funkstation	C: Bar
Frame 5	A: Funkstation	B: Lager	C: Bar

Abbildung 2: Dieses Schema verdeutlicht, wie **Primzahlen im Wechselsystem** unerwünschte Wiederholungen von Objektkombinationen vermeiden.





Klickflächen

Meist benötigen die Programmierer Definitionen für die Klickflächen. Die Anforderungen sind allerdings sehr unterschiedlich: In einem einfachen Browserspiel sind sie normalerweise rechteckig und werden über zwei Bildschirmkoordinaten festgelegt. Bei anderen Spielen werden die Klickflächen aus eigenen Masken ausgelesen, manchmal auch nur aus einer einzigen mehrfarbigen Map.

Die letzten beiden Fälle sind die einfachsten für uns: Wir erstellen einen eigenen Renderlayer oder eine Kopie der Szene. Dann färben wir die entsprechenden Gebäude mit einfarbigen, selbstleuchtenden Materialien ein und entfernen sämtliche Lichter. Das Anti-Aliasing wird ausgeschaltet. Als Ergebnis erhalten wir saubere Klickflächen der Gebäude, die wir unverändert an die Programmierer weitergeben können.

Schwieriger wird es, wenn die Klickflächen rechteckig sein sollen. Die offensichtlichste Möglichkeit ist ein Photoshop-Skript, das die Masken analysiert, Rechtecke erstellt und dabei Überschneidungen vermeidet. Abgesehen von anderen Workarounds ist hier tatsächlich noch Handarbeit praktikabel. Denn die Anzahl der Kamera- und Gebäudepositionen ist begrenzt, alle anderen tauschbaren Elemente sind für die Klickflächen nicht relevant. Bei 13 Gebäudepositionen und 7 Kamerapositionen müssen »nur« 91 Sets an Klickflächen erstellt werden, dann beginnen die Klickflächen wieder von vorn.

Tricks für mehr Abwechslung

- **Mehr Details in die Gebäude einfügen**: Die Gebäude müssen nicht zwingend einfach sein. Wenn sich die Details dynamisch verändern, ist es umso besser. Auch Zwischenstockwerke und zivile Gebäude im gleichen Stil sind möglich.
- **Übergänge verschachteln**: Spieler sind es aus Strategiespielen gewohnt, ein Gebäude neben das andere zu setzen. Einfache Schnitte werden daher schnell erkannt. Stattdessen gestalten wir Übergänge mit überlappender Geometrie, mit Texturverläufen. Oder wir erstellen eigene Objekte (z.B. Technik oder Pflanzen), die die Übergänge dynamisch wechselnd gestalten.
- **Spezialeffekte verwenden**: Tiefenunschärfe, Bewegungsunschärfe, Linseneffekte und Überblendungen können dem Bild mehr Glaubwürdigkeit verleihen. Und man kann sie genauso durchtauschen wie alle anderen Elemente.
- **Lebewesen**: Besonders simpel sind fliegende Vögel in der Entfernung, weil für ihre (dynamische) Positionierung die Umgebung egal ist. Einfache Menschen hinter Glasscheiben oder in der Entfernung wirken auch gut, sind aber aufwendiger zu positionieren.
- **Per Hand nachbearbeiten**: Hier ein wenig Rauch andeuten, dort die Lichter betonen, beim nächsten Bild die Farben etwas aufdrehen. Ein geübter 2D-Artist kann in gut zehn Minuten jedem Bild etwas zusätzliche Individualität verleihen. Das ist natürlich nur dann sinnvoll, wenn man 500 Bilder oder weniger hat.

Die sechste mögliche Kombination (Funkstation – Bar – Lager) ignorieren wir. Es geht nicht darum, zwingend alle möglichen Kombinationen einzusetzen. Wichtiger ist die Abwechslung, und eine Primzahl wie 5 hilft uns dabei mehr als eine komplette Kombinationsabdeckung.

Jede Sichtbarkeit sollte auf dem ersten und fünften Frame einen Key haben. Auch dann, wenn er nicht nötig erscheint. Der Grund: Wir stellen die Animationskurven so ein, dass sie sich ständig wiederholen. Vorsicht bei Maya: Mit der Einstellung »post infinity cycle« wird in den Wiederholungen standardmäßig der erste Frame weggelassen.

Als Nächstes tauschen wir die Varianten der Sockel und Gebäude durch, zu sehen in **Abbildung 3**. Wenn wir beispielsweise von der Bar 3 Varianten haben, dann schalten wir diese einfach nach dem Schema **123-123-123-123 ...** durch. Wenn die Funkstation zufälligerweise auch genau 3 Varianten hat, wiederholen wir einfach die schönste: **1223-1223-1223 ...** Wenn wir noch auf die Primzahl 7 kommen wollen, können wir natürlich auch **1122233-1122233-1122233 ...** wiederholen. Eine Anzahl von 5 sollten wir vermeiden, denn sonst würden (bei **12233**) auf dem Sockel A nie die ersten beiden Varianten der Funkstation zu sehen sein. Alle Animationskurven stellen wir wieder auf Endlos-Wiederholung.

Positionieren wir nun die Gruppen. Wir schieben sie in interessante Zusammenstellungen und setzen für jede einen Keyframe. Weil die Positionierung sehr schnell geht, können wir gleich 19, 23 oder 29 davon setzen. An dieser Stelle können wir einen ersten Test rendern. Von den 200 Frames sollten keine zwei identisch sein.

Texturen und Materialien

Jeder Concept Artist kennt den Trick: Man rendert ein paar einfache Modelle ausschließlich mit Licht und Occlusion, Farbe und Struktur werden erst später in Photoshop hinzugefügt. Das ergibt häufig einen natürlichen Look, spart die Arbeit für die UV-Map und vor allem den Aufwand, jede Textur auf die UV-Map anzupassen.

Würden wir diese Methode 1:1 bei einer automatisierten Generierung benutzen, müssten wir umständlich Masken rendern und Photoshop beibringen, wie diese Masken mit Texturen zu füllen sind. Deutlich einfacher ist es, die Texturen nicht erst in Photoshop, sondern gleich in Maya über die Gebäude zu legen. Eine Kameraprojektion macht's möglich. Für den Vordergrund brauchen wir zunächst zwei Materialiengruppen:

- Oberfläche der Gebäude (neues Plastik, abgeplatzter Lack, Aluminium etc.)
- Metall (Rost, Aluminium, Kupfer, Stahl etc.)

Zum Schluss benötigen wir pro Gruppe ein zentrales Material, in dem die anderen Materialien durchgetauscht werden. Maya-Benutzer erstellen einen Layered Shader und füttern ihn mit den betreffenden Materialien. Die Transparenz-Werte werden (im Layered Shader oder in den Materialien selbst) mit Keyframes animiert, sodass immer nur eines der Materialien vom Layered Shader angezeigt wird. Der Wert des Compositing Flag darf dabei nicht auf »Layer Shaders« stehen, sondern muss überraschenderweise auf »Layer Texture« geändert werden; ansonsten werden zwar die Farben der Materialien, nicht aber ihre Glanzlichter ausgeblendet. Wem die Steuerung aller Transparenzen eines Layered Shaders zu umständlich ist, der kann auch eine »Ramp« erstellen und die Materialien als Farbwerte benutzen. Anschließend muss nur noch der Eingangswert der Ramp-Node animiert werden.

Himmel und Beleuchtung

Kaum etwas beeinflusst eine Szene so stark wie die Lichtgestaltung. Nasskaltes Morgenlicht bei diesigem Himmel? Rotgoldene Abendstimmung? Das sanfte violette Leuchten eines Weltraumnebels oder das überstrahlende Glühen der Mittagshitze? Alles ist machbar – und alles sollte gemacht werden. Weil »mehr« hier auch »besser« bedeutet, ist dies die ideale Gelegenheit, Skyboxes aus alten Projekten zusammenzusuchen. Zu jeder Skybox erstellen wir ein passendes Licht-Setup. Dann kommt die Skybox mit ihren Lichtern in eine Gruppe, die wiederum abwechselnd mit den anderen Beleuchtungsgruppen angezeigt wird. Um die Gebäudebeleuchtung kümmern wir uns später.

Umgebungen

Um die Gebäude nicht im Leeren stehen zu lassen, müssen Umgebungen her. Auch hier lohnt es sich, alte Projekte zu plündern. Wer will, kann auch die Materialien der Hintergründe automa-

Abbildung 3: Hier werden die **Positionen der Sockel** verändert. Auch die Varianten von Silo und Funkturm wechseln. Die Kreise dienen als Orientierung beim Positionieren der Gebäude und bei der Justierung der Kameras.

tisch durchwechseln. So wird etwa aus der obligatorischen grünen Hügellandschaft eine Sandwüste, eine Schneeebene und ein vulkanisches Ödland. Im konkreten Beispiel wurde eine Gruppe von Felstexturen verwendet. Für mechanische Details im Hintergrund verwenden wir natürlich dasselbe Material, das auch an den Gebäuden zu sehen ist. Generell wirken auch farbschwache Materialien gut, weil sie schön auf die jeweilige Lichtfarbe reagieren und so das Gesamtbild harmonischer wirken lassen.

Wir tauschen die Hintergründe in jedem Frame durch – aber wir geben den Umgebungen auch eine leichte Drehung und Verschiebung über mehrere hundert Frames. Denn es irritiert, wenn bei einer bestimmten Umgebung ein auffallender Fels immer an derselben Stelle steht.

Gebäudebeleuchtung

Gebäudebeleuchtung lohnt sich aus drei Gründen: Erstens ermöglicht sie uns Dämmerungs- und Nachtszenen. Zweitens gibt sie dem Betrachter einen Eindruck vom Maßstab, denn beleuchtete Fenster oder Straßenlichter lassen auf die Größe der Gebäude schließen. Drittens gibt uns Gebäudebeleuchtung eine weitere Möglichkeit, die Atmosphäre des Bildes automatisiert zu verändern. Schließlich wirken warme gelbliche Funzeln ganz anders als stahlblaues Neonlicht!

Wir verteilen zuerst rein weiße Lichtquellen in der Szene. Jede dieser Lichtquellen sollte einen linearen oder quadratischen Decay haben – oder zumindest eine begrenzte Reichweite. Fast alle Lichtquellen werden an Gebäuden oder Geländestücken befestigt. Wenn sich die Gebäude durchtauschen, sollten sich die Lichter mitbewegen. Anschließend verknüpfen wir die Helligkeit und die Lichtfarbe der Lichter zu einem zentralen Punkt, etwa dem auffälligsten Licht oder einem eigenen Anfasser. Alle Lichter sollten sich nun zentral dimmen und einfärben lassen. Die Helligkeit der Gebäudebeleuchtung muss parallel zur Skybox bzw. zur globalen Beleuchtung laufen. In Nachtszenen müssen die Lichter voll leuchten, in Abendstimmungen ein wenig, und bei Tagbildern im Prinzip gar nicht.

Für die Farbe der Lichtquellen erstellen wir eine Palette aus Weiß, oben genanntem Neonlicht-Blau und schönen warmen Gelbtönen. Alle anderen Farben sind natürlich auch möglich. Grüntöne erzeugen beispielsweise eine Farbstimmung wie bei den Borg. Die Farbpalette lassen wir mittels Keyframes durchlaufen, fließende Übergänge sind dabei absolut erwünscht. Optional können wir noch einige farbige Lichter verteilen, etwa ein kleines Lämpchen in einem Technikschacht oder ein paar Warnlichter.

Werbung und Schriften

Kleine Hinweise, Warnaufschriften oder Verkehrszeichen gibt es überall. Propagandaplakate und Werbebanner fallen sogar per Definition ins Auge. In unserem Beispiel können wir auf der Bar eine Neontafel anbringen. Die Form der Tafel definieren wir mit einer Schwarz-Weiß-Textur. Je mehr Varianten wir erschaffen, desto besser – ein paar Beispiele sehen Sie in Abbildung 4. Denn sie sind schnell erstellt, haben aber großen Einfluss auf die Individualität des Einzelbildes. Je nach 3D-Programm können wir die UV-Map der Reklame animieren oder den Offset der Textur verändern. Die Farbe der Neontafel animieren wir unabhängig davon.

Rendering und Nachbearbeitung

Grundsätzlich rendern wir die Bilder genau so, wie wir auch eine Animation rendern würden. Die Nachbearbeitung hängt sehr von der zur Verfügung stehenden Software ab: Wer ein Tool wie Digital Fusion oder After Effects hat, kann die Layer für Occlusion, Farbe und andere Effekt-Layer sehr komfortabel zusammenfassen. Wer auf Photoshop beschränkt ist, sollte sich die Render-Layer schon als PSD-Datei zusammengefasst von seiner 3D-Software ausgeben lassen. Ein Ordner voller PSD-Datein ist per Photoshop-Aktion schnell bearbeitet. Doch Hunderte Male je zwei Einzelbilder zu einem einzigen zusammenzufügen, ist nur per Skript möglich.

Aussortieren

Einen Nachteil haben die generierten Bilder: Manche sehen durch reinen Zufall hervorragend aus, andere – ebenso durch Zufall – allerdings nicht. Manche speziellen Kombinationen verdecken sich, haben ungünstige Farbkombinationen, irritierende Lichtverhältnisse oder verursachen wie in Abbildung 5 einen Positionsfehler. Am besten nehmen wir einen schnellen Bildbetrachter mit einem Hotkey für »Löschen«. Wenn wir von 5.000 erzeugten Bildern sowieso »nur« 800 brauchen, können wir auch einfach unsere Favoriten aussuchen.

Was lässt sich noch generieren?

In diesem Tutorial werden nur einzelne Bilder erzeugt. Generell lässt sich aber fast alles generieren: 3D-Objekte und spielbare Levels, Menschen sowie Felsen, Großstädte oder Wälder, Handwaffen ebenso wie Raumkreuzer.

Allerdings reicht simples Durchtauschen meist nicht aus. Es gibt gewisse Regeln, wie sich die Einzelteile zueinander verhalten: Am Steilhang wachsen Bäume anders als im Tal. Bei Menschen hängen Haut, Haarfarbe und sogar Schädelform voneinander ab. Autos fahren auf bestimmten Straßenseiten, Straßen wiederum sind auf bestimmte Weise zwischen den Gebäuden vernetzt. Um eine hohe Qualität zu erreichen, kommt man auf Dauer nicht um eigene Skripte herum. Dann aber sind der Kreativität keine Grenzen mehr gesetzt. **Jochen Winkler**

Abbildung 4: Mithilfe von **exotischen Schriften** und Anregungen der Kollegen hat man innerhalb weniger Minuten Dutzende Neon-Silhouetten parat.

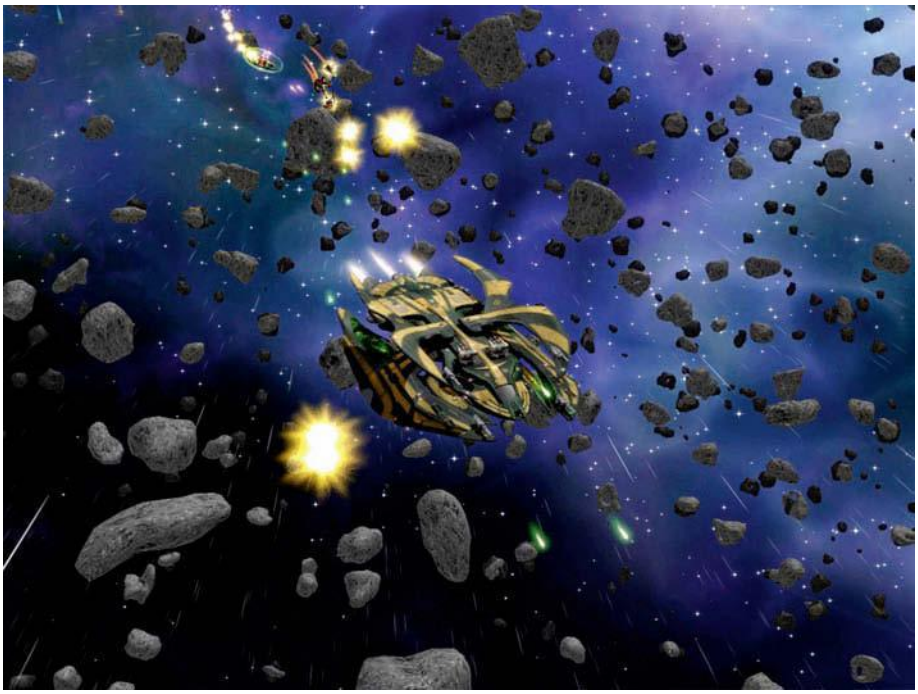
Abbildung 5: Bei dieser **missratenen Asset-Kombination** ist die Kamera unter die Bodenkante abgetaucht. Natürlich hätten wir das schon im Vorfeld vermeiden können, andererseits wären uns dann auch einige spannende Zufallsbilder entgangen.

Upjers

Die Upjers GmbH & Co.KG entwickelt und vertreibt Browserspiele. 2006 von Klaus Schmitt gegründet, zählt Upjers mit Titeln wie »Wurzelimperium«, »My Free Farm« und »Kapi Hospital« zu den führenden Herstellern in der Browsergames-Branche. Die Fokussierung auf Gameplay, Spielgrafiken und Community Management wurde mehrfach durch Awards wie »Browsergame des Jahres« und »Bestes Casual Game« ausgezeichnet.

Spielkonzepte: Workshop Teil 5
Von der Idee zum Konzept

Daniel Dumont erklärt anhand von Darkstar One, warum das theoretische Balancing so wichtig für die Spielkonzeption ist und welche Probleme sich schon frühzeitig mit den richtigen Berechnungen lösen lassen.



In **Darkstar One** steigt das Raumschiff wie ein Rollenspielcharakter im Level auf, neue Waffen inklusive.

Daniel Dumont

ist Mitgründer und Creative Director der Gaming Minds Studios.

Daniel ist seit zwölf Jahren Spieleentwickler. Er entwickelte bei Ascaron die Patrizier-Reihe weiter und entwarf Port Royale und Darkstar One. Im Jahr 2009 gründete er zusammen mit Kay Struve und Kalypso Media die Gaming Minds Studios (Patrizier 4). Sein Schwerpunkt liegt auf der Visionierung sowie auf der Konzeption von Computerspielen.

WWW
makinggames.de
- Teil 1 bis 4 des Konzeptworkshops

Im vorletzten Kapitel der Artikelreihe rund um die Konzepterstellung geht es um das Ausbalancieren von Spielmechaniken – und zwar bereits »vor« der Implementierung. Ich beschreibe eine Methode, wie man komplexe Spielparameter miteinander abgleichen kann, noch bevor man das eigentliche Feature direkt im Spiel testet.

Warum ich häufig so vorgehe, hat drei Gründe:
1. Es gibt viele Balancing-Probleme, die sich nicht so einfach durch Ausprobieren oder Testen lösen lassen.
2. Indem man sich vor der Implementierung (also während der Konzeption) Gedanken über das Balancing eines Features macht, bekommt man eine viel bessere Vorstellung davon, wie das Feature funktionieren soll und von welchen Parametern es beeinflusst wird.
3. Zu einer vollständigen Beschreibung eines Features gehört für mich, dass im Konzept alle Formeln, Parameter und Zusammenhänge beschrieben werden.

Theoretisches Balancing ersetzt natürlich nicht das abschließende Testen im Spiel. Durch gut vorbereitetes Balancing wird jedoch das spätere Testen viel effektiver und man weiß sofort, an welchen Werten man drehen muss, wenn sich das Feature noch nicht wie gewünscht anfühlt.

Das Grundprinzip

Ein Feature zu balancen bedeutet, dass man die enthaltenen Parameter oder Situationen miteinander abgleicht, um ein bestimmtes Spielerlebnis zu ermöglichen. Nehmen wir an, ein Spieler darf in einem Rollenspiel ab Charakterlevel 5 eine bestimmte Höhle betreten, in der er auf einige Gegner trifft und am Ende einen Schatz findet. Dann wollen wir als Game Designer sicherlich nicht, dass er direkt beim Betreten der Höhle von den Gegnern überrannt wird. Wir wollen aber natürlich auch nicht, dass er den Schatz vollkommen mühelos erreicht.

Es geht also darum, dass die Stärke der Gegner (kombiniert aus ihrer Zahl, ihren Waffen und ihrer Verteilung in der Höhle) an die Stärke des Spielers in irgendeiner Form angepasst ist. Dazu könnten wir abschätzen, wie stark der Spieler auf seinem Charakterlevel sein wird und ihm so viele Gegner entgegenstellen, dass alle zusammen in etwa so stark sind wie er selbst – wobei

die Gegner noch einen »Bonus« bekommen, da sie in der Regel nicht alle gleichzeitig angreifen.

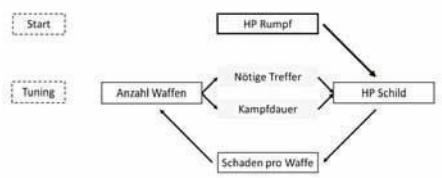
Das Pferd von hinten aufzäumen

Nun könnte man aber auch sagen: Wenn doch die Parameter sowieso miteinander abgeglichen werden, dann kann ich doch genauso gut einen gewünschten Ablauf direkt in die Berechnung einfließen lassen oder gar zusammenhängende Parameter verwenden – immer mit dem Ziel vor Augen, vernünftige Grundwerte für das Konzept zu erhalten.

Dazu ein konkretes Beispiel aus Darkstar One. Hier war es so, dass das Spielerschiff wie ein Rollenspiel-Charakter aufsteigen konnte und auf diese Weise neue Waffenklassen freigeschaltet wurden. Mit jeder dieser Klassen war es dem Spieler möglich, neue Waffen zu kaufen und zu installieren. Gleichzeitig stieg auch die Anzahl der montierbaren Bordwaffen.

Wie musste also die Gegnerstärke zunehmen, damit es der Spieler später nicht zu einfach hatte? Da es sich um ein Actionspiel handelte, galt es außerdem sicherzustellen, dass der Spieler nacheinander mehrere Gegner besiegen konnte und dass ein Kampf nicht zu lange dauerte.

Zu dieser Balancing-Herausforderung zunächst ein kleines Diagramm:



Als Einstiegsparameter für das Balancing habe ich die Hitpoints des Schiffsrumpfes eines Raumjägers in Level 0 genommen, und zwar mit einem willkürlich gewählten Zahlenwert, beispielsweise 100. Ob man nun 1 oder 100 oder 10.000 nimmt, ist wirklich egal, denn nach Abschluss des Balancings kann man die Werte ohnehin noch einmal in Ruhe anpassen.

Aus dem Rumpf konnte daraufhin die Schildstärke ermittelt werden, daraus wiederum der Schaden pro Waffe. Im Anschluss habe ich noch die Anzahl der Waffen eingerechnet und schließlich die nötige Trefferzahl sowie die Kampfdauer abgeleitet.

Der Vorteil dieser Vorgehensweise liegt darin, dass man alle Werte in eine Tabelle einträgt und automatisch alles berechnet wird, sobald man einen Vorgabe-Wert ändert. So kann man mit den Werten herumspielen und erhält – ohne dass auch nur ein Byte Code geschrieben wurde – eine vernünftige Kampfdauer und eine spielerisch sinnvolle Anzahl von Treffern, die ein Raumjäger einstecken darf.

Führt man die Tabelle weiter von Level 0 auf Level 1 und so fort, erhält man außerdem noch eine sinnvolle Steigerung der Werte in Abhängigkeit vom Schiffslevel, also wie sich Rumpf, Schilde, Waffenstärke und die Anzahl der Waffen im Spielverlauf entwickeln.

»Extreme Balancing«

Bevor ich genauer auf das Raumjäger-Beispiel eingehe, möchte ich erläutern, was ich unter »Extreme Balancing« verstehe. Diese Form des Balancings besteht aus zwei Kernpunkten:

Extreme Spielsituationen

Es ist schwierig, sich komplexe Spielsituationen wie beispielsweise einen Raumkampf in allen Details vorzustellen. Das gesamte Geschehen läuft sehr schnell ab und viele unterschiedliche Mechanismen greifen ineinander. Doch in allen Spielsituationen gibt es auch Extremfälle – und genau diese picken wir uns für das Extreme Balancing heraus. Darunter fällt beispielsweise, wenn der Spieler auf einen feindlichen Jäger zufliegt und aus allen Rohren feuert. Oder wenn er einem Gegner im Nacken sitzt und nicht mehr locker lässt. Denn genau solche Extremsituationen sind es, für die sich die gesamte Parameter-Tabelle ideal balancen lässt.

Die Idee dahinter: Wenn die speziellen Situationen, in denen der Spieler ohnehin seinen größten Spaß oder die stärkste Anspannung hat, ausgeglichen sind, dann sind die »normalen« Situationen auch kein Problem für das Balancing.

Extreme Annahmen

Man sollte sich nicht scheuen, von konkreten Annahmen auszugehen, wie das Spiel später ablaufen wird oder wie der Spieler ein Feature erleben soll. Betrachten wir zum Beispiel die Schildstärke unseres Raumjägers: Wie hoch soll sie im Vergleich zur Rumpfstärke sein? Wir beleuchten drei mögliche Fälle:

Fall A: Nehmen wir an, das Schild sei klein gegenüber dem Rumpf. Wenn der Spieler nun in einen zünftigen Kampf gerät, wird das Schild ziemlich schnell weg sein und für den weiteren Verlauf des Kampfes keine große spielerische Bedeutung haben.

Fall B: Nehmen wir an, der Schild-Wert sei groß gegenüber dem Rumpf-Wert. Wenn der Spieler nun in einen harten Kampf gerät und sein Schild verliert, ist sein Schiff bald zerstört. Denn wenn er sogar den starken Schild verloren hat, sind die relativ wenigen Rumpf-HPs natürlich auch bald aufgebraucht.

Fall C: Sind Rumpf und Schild vergleichbar stark, so kann der Verlust des Schildes eine Warnung darstellen und der Spieler hat noch genug Zeit, sich weiter anzustrengen, seine Strategie anzupassen oder aus dem Kampf zu fliehen.

Sinnvoll erscheint hier nur der letzte Fall. Für das theoretische Balancing kann man daher Rumpf und Schild ruhigen Gewissens gleichsetzen und annehmen, dies sei das Beste für das Spielgefühl. Dies ist die »extreme« Annahme. Sie ist ein einfaches Beispiel dafür, dass man nicht an Werten herumdrehen muss, um ein bestimmtes Ergebnis zu erzielen. Manchmal ist es besser, das gewünschte Ergebnis direkt in die Werteberechnung einfließen zu lassen.

Inhalte der Artikelreihe

Teil 1: Die Spielidee (Ausgabe 03/2010)

Teil 2: Der Spielmechanismus (Ausgabe 04/2010)

Teil 3: Das Exposé (Ausgabe 05/2010)

Teil 4: Das Konzept (Ausgabe 06/2010)

Teil 5: Frühes Balancing und die Methode des »Extreme Balancing«

Teil 6: Die Konzeption künstlicher Intelligenz (Ausgabe 02/2011)

Die Berechnung

Nun möchte ich beispielhaft die Erstellung einer Wertetabelle für das obige Diagramm durchgehen. Folgende Werte sollen mithilfe des theoretischen Balancings pro Schiffslevel ermittelt werden:

- Rumpf-HP, Schild-HP, Anzahl Waffen, Waffenstärke
- Steigerung der Werte pro Schiffsstufe
- Zusammenhang zwischen Schiffsstufe und Waffen- bzw. Schildklasse

Wichtig dabei: Je höher die Schiffsstufe, desto stärker wird der Rumpf. Außerdem sollen sich die Waffen- und Schildklassen mit der Schiffsstufe erhöhen, damit der Spieler von Zeit zu Zeit bessere Waffen und Schilde installieren kann. Die Anzahl der Waffen kann ebenfalls an der Waffenklasse hängen. Abbildung 1 zeigt die Stufentabelle für die ersten sechs Schiffsstufen.

Dabei habe ich angenommen, dass sich die Klassen zu Spielbeginn schneller verbessern sollen als später. Außerdem steigen die Waffen- und die Schildklasse nicht gleichzeitig auf, damit der Spieler häufiger etwas zu optimieren hat. In den ersten drei Stufen verbessert sich jeweils eine wichtige Eigenschaft des Schiffes – der Spieler kann sich auf Wunsch also jedes Mal nach neuer Ausrüstung umsehen. Die erweiterte Tabelle sieht nun wie in Abbildung 2 aus.

Erläuterungen zur Balancing-Tabelle

Die Rumpf-HPs sind meine Einstiegsparameter. Genauer gesagt, die bei der Schiffsstufe 0. Hier nehme ich den Wert 100 an und steigere die HPs pro Stufe um 15%. Bei dieser Steigerung kann man sicher sein, dass der Spieler auch den Unterschied bemerkt. Je kleiner dieser Wert, desto weniger spürt er Änderungen. Je höher der Wert, desto größer wird auch der mögliche Kräfte-Unterschied im späteren Spiel (wenn sich beispielsweise zwei Schiffe mit unterschiedlicher Stufe begegnen).

Die Schild-Hitpoints sind gleich den Rumpf-Hitpoints, wobei sich der Schild immer nur dann an den Rumpf anpasst, wenn sich gleichzeitig auch die Schildklasse erhöht.

Nun der Waffenschaden: Ich beginne mit einem Zehntel von Schild+Rumpf. Dadurch ergibt sich ein Schaden von 20 pro Sekunde, der Kampf dauert also mindestens 10 Sekunden (für die »extreme« Annahme, dass ein Schiff das andere permanent im Schussfeld hat). Oder anders ausgedrückt: Wenn meine Waffe eine Frequenz von einem Schuss pro Sekunde hätte, so müsste ich meinen Gegner 10 Mal treffen. Beim Design der Waffen gibt man ihnen natürlich eine unterschiedliche Frequenz, doch der Schaden pro Sekunde wird sich nach dieser Tabelle richten.

Bei den weiteren Waffenklassen habe ich mir erlaubt, die Waffenstärke vom Rumpf zu entkoppeln. Ich gebe den Waffen eine Steigerung von 20%, damit sich die Werte (Rumpf-Schilde-Waffen) asynchron entwickeln. Außerdem wird dadurch die Kampfdauer im Spielverlauf immer kürzer, wodurch wir dem Spieler mehr und mehr Gegner in kürzeren Abständen vor das Fadenkreuz setzen können.

Als Nächstes wird der Gesamtschaden ermittelt. Der ist einfach das Produkt aus Waffenschaden und Anzahl der Waffen. Daraus ergeben sich die Kampfdauer und die Anzahl der nötigen Treffer mit unserer 1-Sekunden-Waffe. Wichtig ist hierbei die Annahme, dass immer alle abgefeuerten Waffen treffen. Da dies in der Realität naturgemäß nicht so ist, wird die Kampfdauer vermutlich höher ausfallen. Doch das spielt keine Rolle, denn wir wollen ja später am fertigen Spiel auch noch Balancing betreiben. Dann führen wir ohnehin noch eine grobe Anpassung der Spielerstärke durch. Hier können wir etwa den Schaden, den der Spieler erhält, mit einem Faktor von 0.5 bis 2.0 multiplizieren, um den Schwierigkeitsgrad und die Gegnerdichte im Gefecht zu berücksichtigen.

Die Tabelle zeigt schon, dass die Kampfdauer abnimmt. Noch einmal zur Erinnerung: Dies ist die Kampfdauer unter der Annahme einer Duell-Situation. Am Anfang, wenn der Spieler das Spiel noch nicht so gut kennt und er nur einem Gegner pro Kampf entgegentritt, ist das durchaus gut so. Im Laufe der Zeit müssen wir aber mehr bieten, um den Spieler bei Laune zu halten. Die Zahl der Gegner erhöht sich daher vielleicht auf durchschnittlich fünf. Der Spieler muss dann etwa während eines Kampfes einen einzelnen Gegner aufs Korn nehmen und sich gleichzeitig vier weitere vom Hals halten. Daher ist es wichtig, dass die Kampfdauer abnimmt, damit sich der Spieler die Gegner schnell nach und nach vornehmen kann. Abbildung 3 zeigt diese Abnahme im Zusammenhang mit der steigenden Spielerstufe. Solche Diagramme sind durchaus sinnvoll beim Balancing, denn

Schiffsstufe	Waffenklasse	Schildklasse	Anzahl Waffen
0	1	1	1
1	2	1	1
2	2	2	1
3	3	2	2
4	3	3	2
5	3	3	2

Abbildung 1: Die **Stufentabelle** der ersten sechs Schiffsstufen aus Darkstar One.

Schiffsstufe	Rumpf-HP	Schild-HP	Schaden/s	Gesamt/s	Treffer & Dauer
0	100	100	20.00	20.00	10.00
1	115	100	28.80	28.80	7.47
2	132	132	28.80	28.80	9.17
3	152	132	34.56	69.12	4.11
4	175	175	34.56	69.12	5.06
5	201	175	34.56	69.12	5.44

Abbildung 2: Die **erweiterte Tabelle** der Schiffsstufen.

man kann überprüfen, ob die Werteentwicklung wie gewünscht verläuft.

Das ganz normale Chaos

Das obige Berechnungsbeispiel ist ein etwas vereinfachter Auszug aus den Berechnungen, die ich für Darkstar One angestellt habe. Es zeigt aber schon eine grundsätzliche Frage auf: Wie sinnvoll ist denn überhaupt die exakte Berechnung einer konkreten Situation, wenn später im Spiel doch sowieso noch weitere Parameter Einfluss nehmen? Dazu einige Überlegungen:

Verbesserung der Konzeption

Für das »Extreme Balancing« betrachte ich zugegebenermaßen Situationen, die im Spiel nur unter vielen Annahmen entstehen. Aber sie kommen vor. Und wenn das Balancing in diesen Situationen nicht funktioniert, wieso sollte es dann in »normalen« Situationen funktionieren?

Man muss in der Konzeption nehmen, was man kriegen kann. Extreme Situationen im Vorfeld zu balancen ist hundert Mal besser als von vornherein nur auf Ausprobieren zu setzen. Und man spart darüber hinaus sehr viel Zeit.

Überlagerung

Die genaue Berechnung von Werten ist gerade deshalb kein Problem, weil später im Spiel das ganz normale Chaos herrscht. Dem Spieler werden die genauen Berechnungen gar nicht auffallen, weil sich so viele Parameter überlagern. Die gegnerischen Schiffe haben beispielsweise nicht alle dieselben Waffen und Schiffstypen. Und der Spieler wird sich nicht strikt an die Annahmen zur Aufrüstung seines Schiffes halten. Dieses Chaos an Werten ist absolut gewollt. Es »verwischt« die Spuren, die unser Balancing ansonsten hinterlassen würde.

Die Richtung zählt

Indem man sich einen bestimmten Spielablauf vorstellt und diesen für das Balancing als Basis annimmt (z.B. Kampfdauer, Anstieg der Waffenstärke), gibt man Entwicklungen im Spiel eine Richtung vor. Selbst wenn man sich zunächst nur auf konkrete Situationen beschränkt, bleibt die Richtung im späteren Spiel erhalten und stützt das weitere Balancing ungemein.

Weitere Anwendungen

Hier noch zwei kurze Beispiele, wie man mit der oben beschriebenen Methode vorgehen könnte:

Produktion/Verbrauch

Bei Systemen, die einen Abgleich von Produktion (Waren, Waffen, Einheiten ...) und Verbrauch (Konsumieren, Verluste im Kampf ...) enthalten, bietet es sich an, rückwärts vorzugehen und erst zu klären, wie hoch der Verbrauch am Ende überhaupt sein soll. Dabei berücksichtigt man unter Zuhilfenahme unterschiedlicher Annahmen die nötigen Vorprodukte und erstellt daraus ein komplettes Bedarfsprofil. Erst anschließend ermittelt man daraus die Produktion.

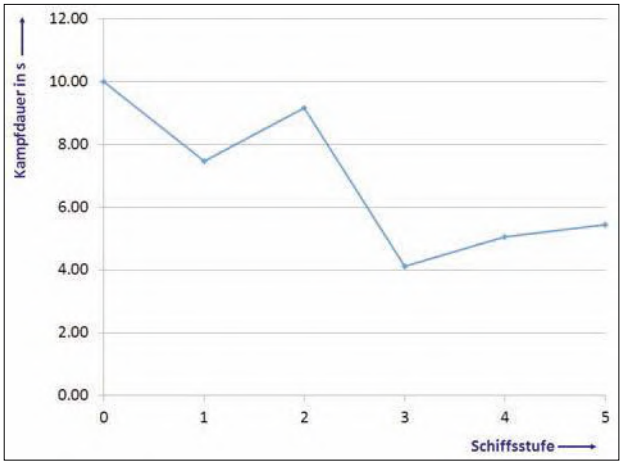


Abbildung 3: Die **Abnahme der Kampfdauer** im Spielverlauf.

Einheiten in einem Strategiespiel

Wenn man viele Einheiten miteinander abzugleichen hat (Schaden pro Sekunde, Hitpoints, Rüstung, Reichweite, Bewegungsgeschwindigkeit ...), kann man versuchen, alle defensiven Fähigkeiten in »Hitpoints« und alle offensiven Fähigkeiten in »Schaden« auszudrücken.

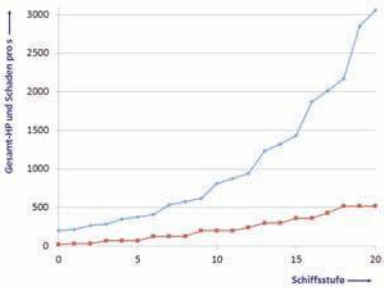
Für abstraktere Dinge wie Reichweite und Fortbewegung stellt man sich konkrete Kampfsituationen vor und ermittelt zusätzliche Größen wie »Erhaltener Schaden, bis Ziel in Reichweite«. So ermittelt man beispielsweise, wie viel Schaden ein Nahkämpfer erhält, bevor er einen Fernkämpfer erreicht hat. Ein Fernkämpfer macht dadurch einen einmaligen, zusätzlichen Schaden, bevor der eigentliche Kampf »Nahkämpfer gegen Fernkämpfer« beginnt. Auf diese Weise kann auch die Bewegung von Einheiten in die Schadensberechnung einbezogen und von vornherein berücksichtigt werden.

Ausblick

Der nächste und letzte Teil dieser Artikelreihe beschreibt, wie sich künstliche Intelligenz – die Programmlogik, die menschliches Verhalten simuliert – konzipieren lässt. Auch hier stellt sich wieder die Frage: Wie lässt sich ein umfangreiches KI-Verhalten so zerlegen, dass dabei leicht verständliche und konzipierbare Häppchen entstehen? Eine Methode hierfür sowie eine Vorgehensweise, wie man sich KI-Problemen grundsätzlich annähern kann, werden dabei ausführlich erläutert.

Daniel Dumont

Sinnvolle Verläufe konzipieren



Bestimmte Werteentwicklungen sind für das Balancing häufig sehr wichtig. Wenn man eine Entwicklung konzipieren muss, sollte man sich immer vorab überlegen, welchen Verlauf sie nehmen soll, damit diese Tatsache gleich in die Berechnung mit einfließen kann. Mit linearen, quadratischen und exponentiellen Verläufen bekommt man eigentlich alles hin.

Interessant am Beispiel Darkstar One ist etwa die Entwicklung der gesamten Hitpoints (blau) und des Gesamtschadens pro Sekunde (rot). Das Schaubild zeigt die Schiffsstufen bis 20, damit man die exponentielle Entwicklung besser erkennt.

XNA-Performancetipps
Parallelisierung für die Xbox 360

Am Beispiel des Xbox 360 Indie Games »Solar Struggle« erklärt Technical Director Oliver Iking, wie Z-Software einen massiven Performance-Schub durch die Parallelisierung in XNA erreicht hat.

Oliver Iking

ist Technical Director bei Z-Software.

Oliver hat 2008 das Entwicklerstudio Z-Software mitbegründet. Seitdem hat er als Technical Director an mehreren Spieleproduktionen mitgewirkt, unter anderem am »Handball-Simulator 2010« und an »Solar Struggle«. Nebenbei leitet er die Entwicklung des Internetportals Partynation.tv und studiert an der TU Dortmund Informatik.

WWW
makinggames.de
- Abenteuer Xbox Live Arcade

Parallelisierung ist heutzutage in aller Munde – nicht zuletzt deswegen, weil sich die Leistungsfähigkeit der heutigen und der zukünftigen Hardware-Plattformen nur dann vollständig nutzen lässt, wenn die Engine ihre Aufgaben an alle CPU-Kerne verteilt. Im Zuge der NextGen-Konsolen und der Multicore-Prozessoren hat rein serielle Programmierung (fast) ausgedient. Normalerweise muss ein Programm außerdem je nach Plattform unterschiedlich optimiert werden. Das XNA-Framework von Microsoft ermöglicht es jedoch, einmal geschriebenen Code sowohl auf der Xbox 360 als auch auf dem PC auszuführen. Dass jedoch gerade dann spezifische Optimierungen vonnöten sind, mag den einen oder anderen überraschen. Wie wir dieses Konzept in »Solar Struggle« und »Solar Struggle: Survival« angewendet haben, ist Thema dieses Artikels.

Das .NET Compact Framework

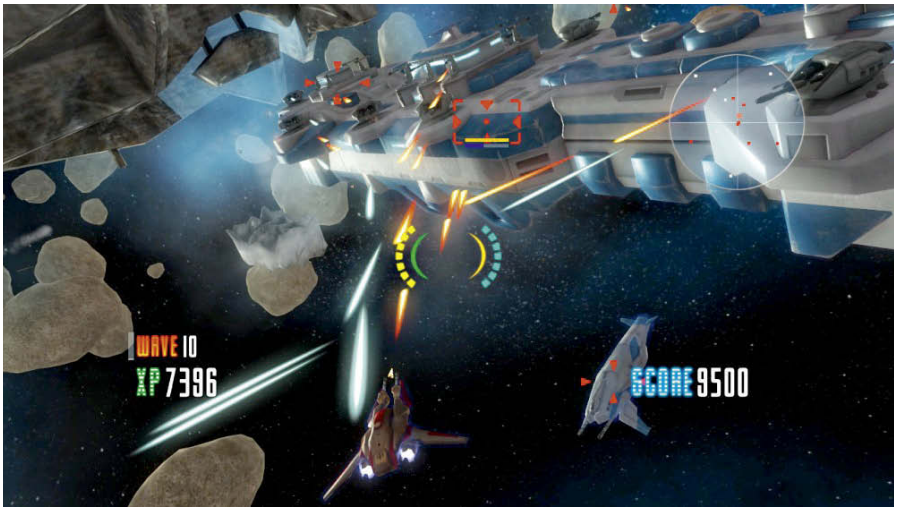
Wer sich schon einmal mit .NET-Sprachen beschäftigt hat, der weiß, dass zum Ausführen von entsprechenden Applikationen das .NET-Framework erforderlich ist. Weil der Programmcode nur in einer Zwischensprache vorliegt – der CIL (Common Intermediate Language) –, kann er nicht direkt auf der CPU ausgeführt werden. Das .NET-Framework sorgt dann zur Laufzeit dafür, dass der CIL-Code in (mehr oder weniger optimierten) Maschinencode übersetzt wird. Zudem muss man sich in .NET-Programmen weniger Sorgen um die Speicherverwaltung machen, weil die Laufzeitumgebung alle Objektinstanzen beobachtet und dabei automatisch erkennt, wann eine Instanz nicht mehr referenziert wird und aus dem Speicher entfernt werden kann.

Weil für diese beiden Aufgaben das Framework zuständig ist, gibt es wesentliche Unterschiede, wenn .NET-Programme auf dem PC oder auf der Xbox 360 ausgeführt werden. Um NET-Programme überhaupt auf der Xbox 360 ausführen zu können, musste sich das .NET-Team von Microsoft eine Reihe von Tricks einfallen lassen, denn die Xbox 360 ist gegen die Ausführung von zur Laufzeit generiertem Code abgesichert. Ein Hauptmerkmal des für Mobile Systems gedachten .NET Compact Frameworks auf der Xbox 360 ist die Zugriffsverweigerung auf die AltiVec-Einheiten. So sind vor allem Vektor, Matrix und Quaternion-Operationen des CF (Compact Framework) auf der Xbox langsamer als auf dem PC. Vektoradditionen benötigen so auf der Xbox drei Additionen statt einer parallel ausgeführten Addition durch AltiVec. Dieser Umstand führt dann in berechnungsintensiven Szenarien zu Performance-Problemen.

Ein weiterer Punkt, der zu einem Bottleneck werden kann, sind übermäßig viele (virtuelle) Funktionsaufrufe. Um dieses Problem anzugehen, sind jedoch andere Optimierungsansätze nötig. In Bezug auf das Kollisionssystem war die Optimierung mittels Parallelisierung schon ausreichend, um den Fokus danach wieder auf andere Teilsysteme wie das Rendering oder die allgemeine Spiellogik zu verschieben.

Kollisionsprobleme

In unserem aktuellen Projekt »Solar Struggle« sind diese Probleme im Kollisionssystem aufgetreten. Solar Struggle ist ein typischer Weltraum-Shooter; durch die Arcade-Ausrichtung geizt er nicht mit Effekten. Vor allem die Waffensysteme spucken eine Menge Energieprojektile aus, sodass in einer typischen Spielsituation



Im Weltraum-Shooter **Solar Struggle** besitzen Großkampfschiffe sehr viele Auto Cannons und sind aufgrund ihres hohen Projektil-Ausstoßes ein kritischer Moment für das Kollisionssystem.

schon einmal mehrere hundert Projektile gleichzeitig berechnet werden müssen. Das Kollisionssystem in Solar Struggle war dabei anfangs relativ einfach gestrickt und erlaubte daher nur Kreis/Kreis-Kollisionen.

Im späteren Projektverlauf haben wir allerdings noch konvexe Polygone und Liniensegmente als Kollisionsprimitive hinzugefügt, um die Kollisionsgenauigkeit von Großkampfschiffen und Projektilen zu verbessern. Eine erste Optimierung war es, Kollisionsgruppen einzuführen, um sinnlose Tests zu vermeiden. Weil es vom Spieldesign nicht gefordert war, schlossen wir zunächst Kollisionen von Projektilen untereinander aus. Dies gab schon einen ordentlichen Performance-Schub, weil die Projektile einen Großteil der Kollisionsobjekte ausmachten. Zusätzlich wurden auch Kollisionen unter statischen Objekten (z.B. unbewegliche Asteroiden oder Schutt) von vornherein ausgeschlossen.

Die kritischste Komponente beim Design eines Kollisionssystems ist die sinnvolle Unterteilung des Raumes, in dem Kollisionen stattfinden können. Ohne eine hierarchische Unterteilung würde das Problem eine quadratische Laufzeit besitzen. Folglich war die zweite – etwas komplexere – Optimierung, den Kollisionsraum in ein gleichmäßiges Grid aufzuteilen und nur noch Kollisionen zu betrachten, die innerhalb der einzelnen Zellen auftreten können. Dies gab zwar einen weiteren Geschwindigkeitsschub, doch weil die Kämpfe in Solar Struggle meist auf einem eng begrenzten Gebiet stattfanden und Projektile in sehr kurzen Abständen abgefeuert wurden, führte dies zu einer ungleichen Verteilung innerhalb der Kollisionszellen: Einige waren sehr gering bevölkert (z.B. am Spielfeldrand), andere hingegen übermäßig (z.B. die Zelle, in dem sich der Spieler gerade befand). So haben wir als nächsten Schritt die Algorithmen zum Testen der Projektil/Spielobjekt- und der Spielobjekt/Spielobjekt-Kollisionen voneinander getrennt, um danach diese beiden Berechnungen parallel ausführen zu können.

Parallelisierung

Um eine Basis für weitere Paralellisierungsmöglichkeiten zu besitzen, haben wir damit begonnen, ein passendes Framework für das parallele Abarbeiten von Aufgaben zu schreiben. Unser gewählter Ansatz betrachtet unterschiedliche Arbeitseinheiten, sogenannte »Jobs«, und ihre Abhängigkeiten untereinander. Wesentliche Eigenschaft des Job-Systems ist eine weitestgehend primitive Implementierung, sodass beim Ausführen vieler kleiner Jobs das Management nicht zum Flaschenhals wird. Das System arbeitet nach einem FIFO-Ansatz (First In, First Out): Bei der Initialisierung des Spiels wird einmal ein an die Plattform (z.B. an die Anzahl der Prozessoren) angepasster Graph von Jobs registriert. Wenn das System nun angewiesen wird, den Job-Graph auszuführen, werden solange Jobs ohne offene Abhängigkeiten an Worker-Threads zur Abarbeitung zugewiesen, bis sie

Bis der Spieler in solchen **Kampfszenen** sein Schiff flüssig steuern konnte, war eine Menge Arbeit nötig.

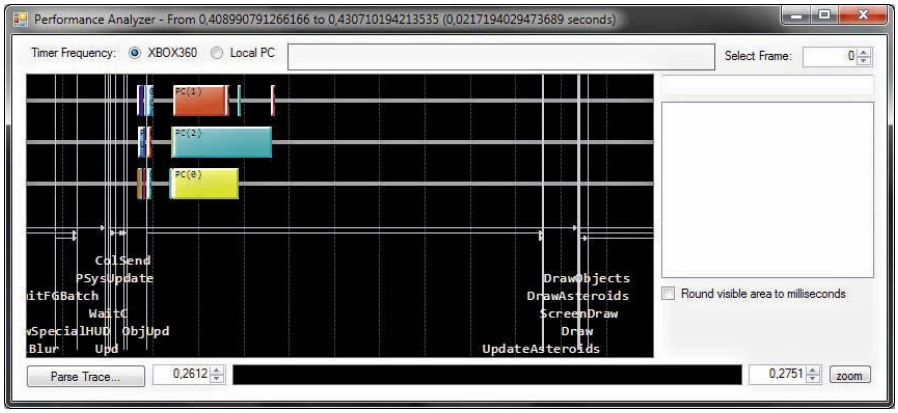
alle ausgeführt wurden. Dies ist ein wichtiger Zeitpunkt, weil die Spiellogik auf die Ergebnisse des Kollisionssystems warten muss, um die betreffenden Spielobjekte über ihre Kollision zu informieren. Die eigentliche parallele Ausführung mehrerer Jobs geschieht durch mehrere Worker-Threads, die jeweils einem Hardware-Prozessor zugeteilt sind. Auf der Xbox 360 ist es dabei zwingend notwendig, die Core-Affinität der Threads von Hand festzulegen, weil der Xbox-Kernel die Threads nicht automatisch verteilt. Auf einem PC ist dies dagegen meist nicht vonnöten, da der Betriebssystem-Kernel die vorhandenen Threads immer bestmöglich auf die vorhandenen CPUs verteilt. Da aber allein das XNA-Framework auf der Xbox 360 schon zwei der sechs Hardware-Threads für sich selbst beansprucht, bleiben für das Spiel nur noch vier Threads übrig. Auf einem läuft der Spiel-(Haupt-)Prozess, folglich liegen lediglich drei Hardware-Threads brach, die wir für die Abarbeitung der Jobs nutzen können.

Debugging & Profiling

Die Performanz des gesamten Systems hängt natürlich maßgeblich davon ab, wie die Aufgaben auf die Worker-Threads verteilt werden und wie sich das System an Synchronisierungspunkten verhält. Dazu hatten wir ein Tool entwickelt, mit dessen Hilfe wir die zeitliche und räumliche Abfolge des Job-Systems analysieren konnten. Dies hat es uns erlaubt, zum einen die Jobs mit dem größten Optimierungspotenzial zu erkennen und zum anderen auch Fehler im Scheduler zu beheben. Um die Debug-Daten von der Xbox zu bekommen, greift man am besten auf die Trace-API zurück. Diese leitet beliebigen Text an das Ausgabefenster vom Debugger und ermöglicht so einen relativ simplen Datentransfer von der Xbox zum PC. Jedoch ist die Trace-API sehr langsam – ein paar Zeilen Trace-Ausgabe können schon einmal eine Sekunde dauern. Daher ist der direkte

Über Z-Software

Z-Software wurde im Jahr 2008 gegründet und hat sich seitdem auf die Entwicklung von Videospielen und das Angebot von Motion-Capture-Dienstleistungen spezialisiert. Darüber hinaus engagiert sich Z-Software mit der Organisation des Branchen-Events »Spieleentwicklertreff.NRW«, um die Vernetzung der Nordrhein-Westfälischen Entwicklerszene zu fördern.
http://www.set-nrw.de/
http://www.z-software.net/



Ein Screenshot aus unserem **Profiler-Analysetool**. Das Programm zeigt Jobs (oberer Bereich) und Events (unterer Bereich). Events besitzen immer eine Start- und Endzeit und werden hierarchisch dargestellt.

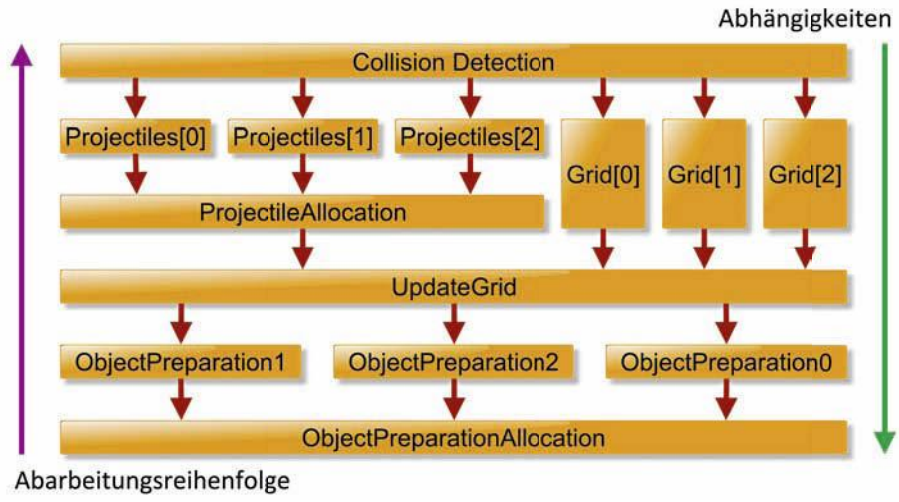
Gebrauch in hochfrequentem Code nicht zu empfehlen, geschweige denn während einer Profiling-Session. Aus diesem Grund legt der Profiler alle Samples in einem Zwischenspeicher ab und streamt sie bei Aufforderung als einfaches Textformat über die Trace-API zu Visual Studio. So lässt sich der Overhead des Profilers möglichst gering halten und die Ergebnisse werden nicht so sehr verfälscht. Im Laufe der Entwicklung haben wir unser Profiling-Tool immer weiter optimiert und nun bietet es auch die »üblichen« Features eines Profilers wie etwa die Messung von einzelnen Funktionsaufrufen. So konnten wir weitere Hotspots erkennen, die sich auf dem PC mittels Slim-Tune aufgrund der veränderten Laufzeitcharakteristik nicht ausmachen ließen.

Portierung

Nachdem das Job-System implementiert war, mussten die einzelnen Schritte des Kollisionssystems zu Jobs zusammengefasst werden. Dies war ein wichtiger Schritt, denn er entschied darüber, wie viel der eigentlichen Arbeit wirklich parallel ausgeführt werden konnte. Dazu haben wir die einzelnen Arbeitsschritte des Kollisionssystems analysiert und angepasst, um später durch Aufteilung der Daten eine parallele Abarbeitung zu ermöglichen. Folgende Arbeitsschritte wurden identifiziert:

- Synchronisieren des Kollisions- mit dem Objektsystem
- Einsortieren aller Kollisionsobjekte in Zellen
- Ausführen der Objekt/Objekt-Kollisionstests
- ausführen der Projektil/Objekt-Kollisionstests

Diese Elemente galt es nun in Jobs zu verpacken. Der erste Schritt war das Synchronisieren des Kollisionssystems. Es aktualisiert die Transformation der Kollisionsprimitive. So wird sichergestellt, dass sich zu Beginn der Kollisionsüberprüfung alle Primitive auf denselben Positionen wie ihre Eigentümer (die Spielobjekte, z.B. Schiff oder Asteroiden) befinden. Der nächste Schritt sortiert die Primitive dann in die Zellen ein, und die letzten beiden Schritte führen die eigentliche Kollisionserkennung durch. Während der Kollisionserkennung werden Kollisionsereignisse generiert, die später von der Spielschleife an die Spielobjekte weitergeleitet werden.



Dieser Graph zeigt die **Abhängigkeiten der einzelnen Jobs**, die für die Kollisionsberechnung (bei drei Worker-Threads) zuständig sind. Die Pfeile geben dabei die Richtung der Abhängigkeiten an, sodass die Jobs in der Reihenfolge von unten nach oben abgearbeitet werden.

Skalierung

Weil die einzelnen Schritte jetzt feststanden, war es einfach, die Beziehungen festzulegen. So konnte die Einsortierung in die Zellen erst vorgenommen werden, als alle Kollisionsprimitive auf einem aktuellen Stand waren. Die Kollisionserkennung war wiederum abhängig davon, ob die Zellen gefüllt waren. Da die Projektil- und die Objektkollisionserkennung nicht voneinander abhängig waren, war die Separation dieser beiden Aufgaben der erste Schritt zur Parallelisierung. Um die Worker-Threads des Job-Systems vollständig auszulasten, mussten aber immer mindestens drei Jobs parallel ausgeführt werden. Weil einzelne Kollisionstests auch nicht voneinander abhängig waren, wurde die Datenmenge der Projekt- und der Objekttests jeweils auf drei Jobs aufgeteilt, die ein Drittel der Arbeit verrichteten. Die Objekttests stellten hier eine Besonderheit dar, weil nicht jeder Job ein Drittel aller Kollisionszellen erhielt, sondern alle drei Jobs sich so lange eine neue Zelle zum Abarbeiten vom Kollisionssystem holten, bis alle Zellen überprüft waren. Die Jobs zum Überprüfen der Projektile erhielten jedoch jeder ein Drittel der existierenden Projektile, um diese zu testen. So ergab sich eine ausreichende Parallelisierung, um den Detailgrad (sprich die Anzahl der Projektile und Objekte) im Spiel weiter zu erhöhen, ohne an eine Performance-Grenze zu stoßen.

Weitere Optimierungsperspektiven

In Zukunft gilt es, die Parallelisierung weiter auszubauen und in der gesamten Engine zu verwenden. Einmal abgesehen von Elementen wie der Physik gibt es noch eine Menge Bereiche, die durch eine nebenläufige Ausführung sowohl die Spiel-Performance als auch die Möglichkeiten

der Game Designer auf einen anderen Level heben würden. Denkbar wäre, die KI-Berechnungen ebenfalls parallel zur eigentlichen Spiellogik laufen zu lassen, weil die KI nicht unbedingt auf einen »exakten« Spielzustand angewiesen ist, um Entscheidungen zu treffen. Empfindliche Algorithmen wie Pathfinding oder Planning könnten wir außerdem in Jobs auslagern und die Ergebnisse erst nach deren Ausführung anwenden.

Die Parallelisierung der eigentlichen Spiellogik ist ein weiterer Schritt. Dort ist es jedoch schwieriger, das gewohnte serielle Paradigma beizubehalten. Es sind also neue Ansätze gefordert, weil bei der parallelen Aktualisierung mehrerer Spielobjekte die Abhängigkeiten unter den einzelnen Spielobjekten beachtet werden müssen. Ansonsten läuft man Gefahr, durch Race Conditions einen inkonsistenten Spielzustand zu erzeugen. So könnte ein Spieler-Charakter versuchen, eine Waffe abzufeuern, bei der im selben Moment der Nachladestatus berechnet wird. Ausgeweitet auf ein vorhersehbares Verhalten, wie es in den meisten Multiplayer-Spielen gefordert ist, kann dies einen erheblichen Einfluss auf den Ablauf des Spielgeschehens haben. Daher wird in dieser Richtung auch einiges an Forschung betrieben. Bereits jetzt gibt es einige Ansätze, wie man das Problem lösen könnte. Unter anderem sind etwa Ideen wie Message-Passing im Gespräch, bei dem Spielobjekte nur noch über Nachrichten untereinander kommunizieren; oder Double-Buffering, bei dem während des Spielzustand-Updates nur auf die Spielobjekt-Zustände des letzten Frames zurückgegriffen werden darf.

Einige dieser Ideen stammen aus dem MMO-Genre, wo die Spielwelt schon aufgrund des technischen Standpunktes parallel auf mehreren physikalischen Maschinen simuliert werden muss. Abschließend darf auch die Parallelisierung des Renderers nicht außer Acht gelassen werden. Mit DirectX 11 hat Microsoft seine Grafik-Schnittstelle endlich (echt) Multithreading-fähig gemacht, und man darf gespannt sein, wie Grafik-Engines diesen Vorteil in Zukunft nutzen werden.

Fazit

Es ist anfangs ziemlich ernüchternd, wenn man die Performance seines XNA-Spiels zwischen PC und Xbox vergleicht. Nach der Parallelisierung des Kollisionscodes ist dieser jedoch wieder in den Hintergrund gerückt und musste in der weiteren Entwicklungszeit nicht mehr angefasst werden. Mit den richtigen Techniken bekommt man so die Flaschenhälse in den Griff und kann sein Spiel sowohl für den PC als auch für die Xbox 360 massiv optimieren. **Oliver Iking**

Solar Struggle: Survival

»Solar Struggle: Survival« ist eine Ausgliederung des Xbox Live Indie Games »Solar Struggle«. Im Gegensatz zum Hauptspiel besteht »Solar Struggle: Survival« nur aus dem Freispielmodus, der um zusätzliche Powerups und abwechslungsreich inszenierte Bosskämpfe erweitert wurde.

Testdokumentation, Teil 1
Test Plan & Test Case

In unserer Artikelreihe beschreibt Keyfactor-Mitgründer Frank Fay, wo die Anforderungen an eine Testdokumentation für die Qualitätssicherung liegen. Im ersten Teil steht das Erstellen von Test Plan und Test Case im Fokus.

Frank Fay

ist Mitgründer und Production Director bei Keyfactor Entertainment.

Frank ist seit 1995 in der Games-Branche tätig und arbeitete zuvor im In- und Ausland in der Qualitätssicherung, der Produktakquise, im Marketing und der Entwicklung bei führenden Unternehmen wie Atari, Infogrames, Microprose, Psygnosis und Sunflowers. Er ist zertifizierter Project Management Professional (PMP), Prince2 Practitioner, Certified Scrum Master (CSM) und Projektleiter (IHK).

Neben Projektleitung, Programmierung, Grafikerstellung, Game Design und Lokalisierung ist das Testen ein fester Bestandteil des Entwicklungsprozesses. Das Testteam plant den Testprozess und bestimmt, welche Bereiche mit welchen Tools und Techniken überprüft werden sowie nach welchen Regeln das Testen abläuft. Die Erstellung einer Testdokumentation ist damit eine essenziell wichtige Aufgabe zur Qualitätssicherung von Computerspielen.

Dieser Workshop erklärt, wie eine gute Testdokumentation aussehen muss und wie sie erstellt wird. Sämtliche verwendeten Beispiele stammen aus der Praxis und sind deshalb in englischer Sprache verfasst. Vorkenntnisse im Software Testing und Projektmanagement erleichtern das Verständnis.

Warum ist eine Testdokumentation wichtig?

Viele Projekte verfügen meist über eine oder mehrere der folgenden Eigenschaften:

- Sie entstehen unter Zeitdruck.
- Spezifikationen ändern sich mit der Zeit.
- Funktionalitäten von unfertigen Komponenten müssen getestet werden.

Für das Testteam bedeutet dies, dass das Testen genau geplant und mit der Entwicklung synchronisiert werden muss. Planen bedeutet daher auch, die einzelnen Schritte genau zu dokumentieren und die Dokumentation über den gesamten Projektverlauf hinweg zu pflegen.

Unternehmens- und projektrelevante Dokumentation

Es ist wichtig hervorzuheben, dass es zwei Arten der Dokumentation zu unterscheiden gilt. Beide müssen hierarchisch verstanden werden:

1. Unternehmensrelevante Dokumente
2. Projektrelevante Dokumente

Unternehmensrelevante Dokumente werden projektübergreifend verwendet. Darin werden Unternehmensprozesse, -standards, -Guidelines usw. festgehalten.

Im Bereich des Qualitätsmanagements würden zum Beispiel folgende Informationen dokumentiert:

- Job Descriptions
- Rekrutierungsprozess
- Schulungsprozesse
- Performance Review Prozess

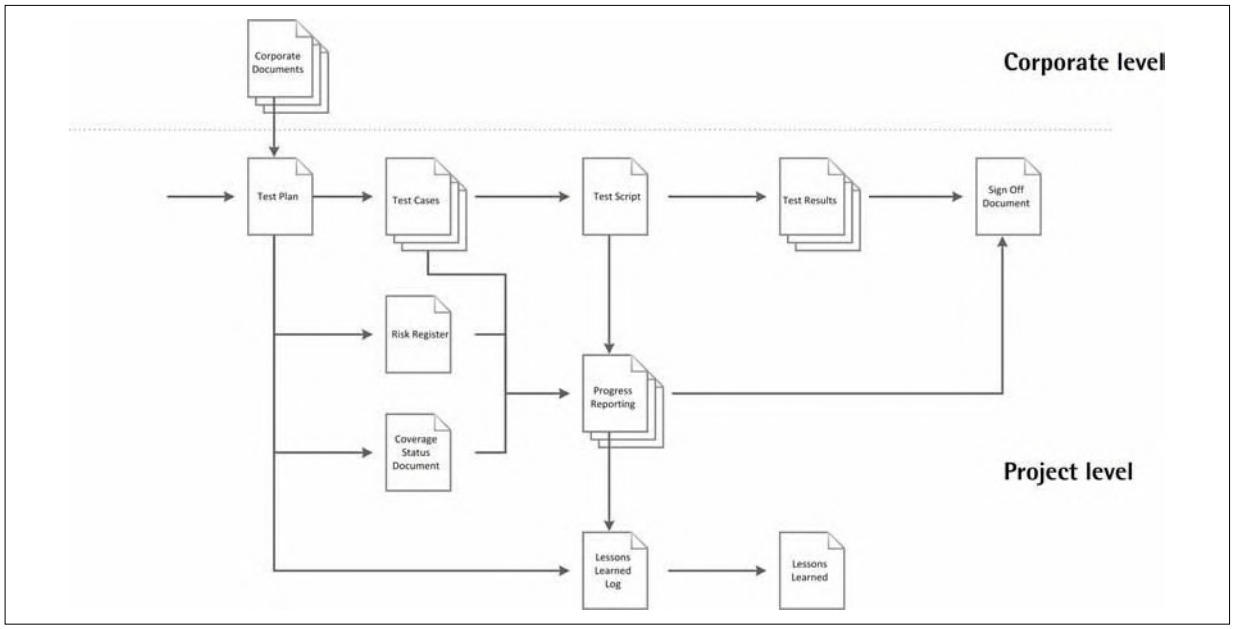


Abbildung 1: Das Flowchart zeigt die **zwei Dokumentationsebenen** des Testprozesses.

- Milestone Sign Off Prozess
- Qualitätsmanagement Plan
- Reportingprozess
- Bug Writing Guidelines
- Bug Definitions
- Sicherheitsregeln
- usw.

Weil jedes Unternehmen anders aufgestellt und strukturiert ist, ist der Gegenstand dieses Workshops die Dokumentation auf Projektebene. Die projektrelevante Dokumentation beinhaltet auf das Projekt bezogene Informationen bzw. nur für das Projekt geltende Abläufe, Prozesse, Metriken usw. Die Beziehungen zwischen den einzelnen Dokumentationen im Testprozess sehen Sie in **Abbildung 1**. Wenn unternehmensrelevante Standards zur Geltung kommen, wird in diesen Dokumenten darauf verwiesen und eventuelle Ausnahmen/Änderungen werden geltend gemacht.

Generelle Inhalte

Jedes Testdokument sollte über ein einheitliches Layout, eine einheitliche Struktur verfügen, soweit es die verwendete Software zulässt. Ferner sollten folgende wiederkehrende Elemente in allen Dokumenten verwendet werden:

- Project Details
- Document Identifier
- Version Number
- Date of Modification
- Document Objectives
- Author
- Date of Creation
- Change Log

Project Details

Hier sollten der Name des Projekts und die entsprechende(n) Plattform(en) stehen.

Document Identifier

Um ein Dokument im Projektverlauf oder im Nachhinein zuordnen bzw. auffinden zu können, sollte es über einen spezifischen Namen verfügen, der gleichzeitig auch der Filename ist. Beispiel: Test Plan-ProjektX-v2-23-09-2010

Author

Der oder die Verfasser des Dokuments sollten immer bekannt sein, um zum Beispiel Rückfragen stellen zu können. Es ist auch empfehlenswert, auf Namenskürzel zu verzichten.

Version Number

Es sollte immer eine Versionsnummer angegeben werden, auf die man sich in der Kommunikation beziehen kann. Sie dient auch der Identifikation der aktuellsten Version.

Date of Creation/Date of Modification

Anhand der Datumsangaben lässt sich das Dokument chronologisch besser einordnen.

Change Log

Hier werden die Änderungen seit der letzten Dokumentversion nach Datum protokolliert. Ein Beispiel:

23/07/2010 **Version 2.0**

Scope

• Added Localisation Testing Description
• Removed several typos

14/09/2010 **Version 3.0**

Quality standards

• Added Bug Writing Guidelines

Document Objectives

Der Sinn und Zweck des Dokuments sollte eingangs erwähnt werden, damit die darin enthaltenen Informationen im Kontext stehen. Diese Sektion sollte fester Bestandteil des Templates sein und auf Unternehmensebene, zum Beispiel vom Project Management Office (PMO) erstellt werden. Beispiel:

The purpose of the Test Plan (TP) is to provide a broad overall framework for implementing quality management to ensure the project's successful execution. The audience is the project stakeholders and the project team members.

The TP will outline the quality control (QC) and quality assurance (QA) for the project to provide a foundation on which the project deliverables will meet the expectations of the project.

The TP will outline the various roles and responsibilities of the team in managing the project's quality processes and ensuring they are implemented and followed.

The goals of the Quality Management are:

1. *to describe the project quality management activities that are planned.*
2. *to define measurable quality standards and their priorities for the project.*
3. *to ensure that the project progress in achieving the quality standards is quantified and managed.*

Test Plan

Der Test Plan beschreibt die zu verwendenden Qualitätstechniken und -standards und legt die Zuständigkeiten fest. Er definiert die Teststrategie des Projekts, um die geforderte Qualität zu erreichen. Folgende Punkte werden in einem Test Plan beschrieben:

- Scope
- Quality Tools
- Project Organization Structure
- Testing Strategy
- Build Delivery / Deployment Schedule
- Reporting
- Communication
- Purchasing Plan
- Quality Standards
- Acceptance Criteria
- Roles and Responsibilities
- Test Schedule
- Test Area and Resource Breakdown
- Escalation
- Training Plan

Inhalte der Artikelreihe

Teil 1: Übersicht, Test Plan, Test Case (01/2011)
Teil 2: Test Script, Risk Register, Coverage Doc, Lessons Learned Log (02/2011)
Teil 3: Progress Reporting, Test Results, Lessons Learned, Sign Off Document (03/2011)
Teil 4: Dokumentation in der Praxis (04/2011)

Keyfactor Entertainment

Keyfactor Entertainment ist eines der führenden Dienstleistungsunternehmen für Qualitätssicherung, Evaluation, Produktion und Kommunikation von Computer- und Videospielen. Die 2004 gegründete Firma beschäftigt heute über 80 Mitarbeiter aus elf Nationen. Seit Bestehen des Unternehmens wurden mehr als 1.000 Projekte für alle relevanten Plattformen getestet.
www.keyfactor-entertainment.com

Scope

Der Scope oder »Statement of Work« beschreibt die Aufgaben, die das Testteam zu erfüllen hat. Der hier definierte Scope hat Einfluss auf die weiteren Sektionen des Dokuments und sollte so genau wie möglich formuliert sein. Wenn sich der Scope ändert, muss das Dokument überarbeitet werden. Es empfiehlt sich auch, die nicht geplanten/geforderten Tätigkeiten zu erwähnen. Beispiel:

The test team will perform the following tasks:
- providing early gameplay feedback
- balancing of the game
- testing the game's functionality
- functional compliance testing

As per verbal agreement, the test team does not perform the following tasks:
- localisation testing
- hardware compatibility testing

Quality Standards

Hier wird definiert, ob das Projekt gewissen Qualitätsstandards folgen soll, die das Unternehmen zum Beispiel in einem Quality Management Plan vorschreibt (siehe unternehmensrelevante Dokumente). Es ist auch möglich, dass das Team spezielle Standards für dieses Projekt entwickeln muss. Beispiel:

The following quality standards will apply to this project:
- bug count definition for signing-off console projects
- bug writing guidelines
- bug classification definitions
- final technical certification guidelines
- Sony and Microsoft compliance guidelines
- closing and waiving procedure

Additionally, quality standards have to be defined for:
- min. frame rate definitions

Quality Tools

In diesem Abschnitt werden alle Tools gelistet, die für das Projekt verwendet werden, darunter Spreadsheets, Templates oder Software Tools.

We will use the following tools for this project:
- Templates
  - Standard Test Case Template
  - Standard Reporting Template
  - etc.
- Software
  - DevTest for managing the Compliance Test Cases
  - DevTrack as bug database
  - Sharepoint for document management
  - InfoPath for reporting
  - etc.
- Spreadsheets
  - Resource allocation matrix
  - Risk register

Acceptance Criteria

Mit dem Entwicklungsteam und dem Auftraggeber/Management gilt es, Abnahmekriterien vor Projektstart festzulegen. Hier wird definiert, welche Kriterien einzelne Funktionen oder das gesamte Projekt erfüllen müssen, um eine Abnahme durch den Auftraggeber/das Management erfolgreich abschließen zu können. Werden diese Kriterien erreicht, dann kann davon ausgegangen werden, dass der Auftraggeber/das Management mit dem Produkt zufrieden ist. Ein weiterer Vorteil ist, dass durch das Festlegen der Kriterien die Erwartungen des Auftraggebers/Managements besser erfüllt werden können und es keine Unklarheiten gibt. Beispiel:

A feature is functional and complete, when:
- the result of the specific test case is as expected.
- the specific test case has been marked as passed.
- the regression of that feature was successful, in case it was previously failed.
- when no open bug concerning this feature is present in the bug database.

The project is accepted as final and complete, when:
- there are 0 open known A bugs in the bug database.
- there are 0 open known B bugs in the bug database.
- there are 0 open required compliance issues in the bug database.
- there are 0 open legal issues in the bug database.

Project Organisation Structure

Für jeden Beteiligten ist es wichtig zu wissen, welcher Ansprechpartner welche Aufgabe im Projekt betreut und wie die Beziehung zu den einzelnen Stakeholdern ist. In der Regel ist dies ein Organigramm, das nur die wirklichen Schnittstellen zur Qualitätssicherung aufweist. **Abbildung 2** zeigt ein Beispiel für eine einfache Project Organisation Structure mit wenigen Stakeholdern.

Roles and Responsibilities

In einem Testteam gibt es unterschiedliche Positionen und Aufgaben, die die einzelnen Teammitglieder wahrnehmen. Jeder von ihnen sollte seine Aufgabe im Projekt kennen und ausfüllen sowie über die der anderen Bescheid wissen (**Abbildung 3**).

Testing Strategy

In diesem Abschnitt wird ausführlich beschrieben, nach welcher Strategie vorgegangen wird, um das Projekt zu testen. Dabei wird auf die Testebenen und Techniken eingegangen. Hier ein Ausschnitt aus einer Testing Strategy:

The functionality team will be able to cover the following aspects of the Functionality Testing:

- Functionality
- Balancing
- Gameplay
- Usability
- Narrative
- Premium Content
- Performance
- Graphics

The general testing direction would be to start each cycle with one full functionality pass starting with the basic game elements (GUI, controls etc.) and proceeding to the gameplay elements that build upon them, with missions being the bigger area to test in the game completion. We will work with test cases and an explorative testing technique.

During this time we will get an overview of the balancing which will allow us to prepare for the balancing tests, which need to be coordinated with the developer.

Any leftover time will be used to test for general functionality issues, bug regression and for providing feedback as desired by the developer. Due to the nature of the game design (number of combinations and random factors) we will not be able to guarantee exhaustive testing on all areas, predominately these areas:

- **Items:** Regarding the large number of items and modifications, we will be unable to check all possible combinations. We'll focus on weapons, and the most useful/needed items. It would be helpful to provide us with a list of all available items and modifications in order to better estimate the work to be done by the FT team in this area.
- **Upgrades:** Can't be fully tested as there are lots of combinations. As above, a list of the different ships and upgrades could be useful for workload estimation.

We will try to cover these areas with Equivalence Class Testing, assuming that a feature will function the same way in similar game locations, and Boundary Value Testing for testing values on either side of edges. Additionally we may identify issues during Explorative Testing Sessions. [...]

Test Schedule

Hier wird die zeitliche Abfolge der Tätigkeiten definiert. Das kann auf Wochen oder Tage heruntergebrochen werden (Abbildung 4).

Build Delivery / Deployment Schedule

Weil die Testaktivitäten abhängig von der Verfügbarkeit der Builds durchgeführt werden, müssen die Build-Lieferungen in die Planung aufgenommen werden (Abbildung 5).

Test Area and Resource Breakdown

Hier wird beschrieben, was zu welchem Zeitpunkt und von wie vielen Testern bearbeitet wird. Anhand dieser Aufstellung werden später auch die Kosten abgeleitet (Abbildung 6).

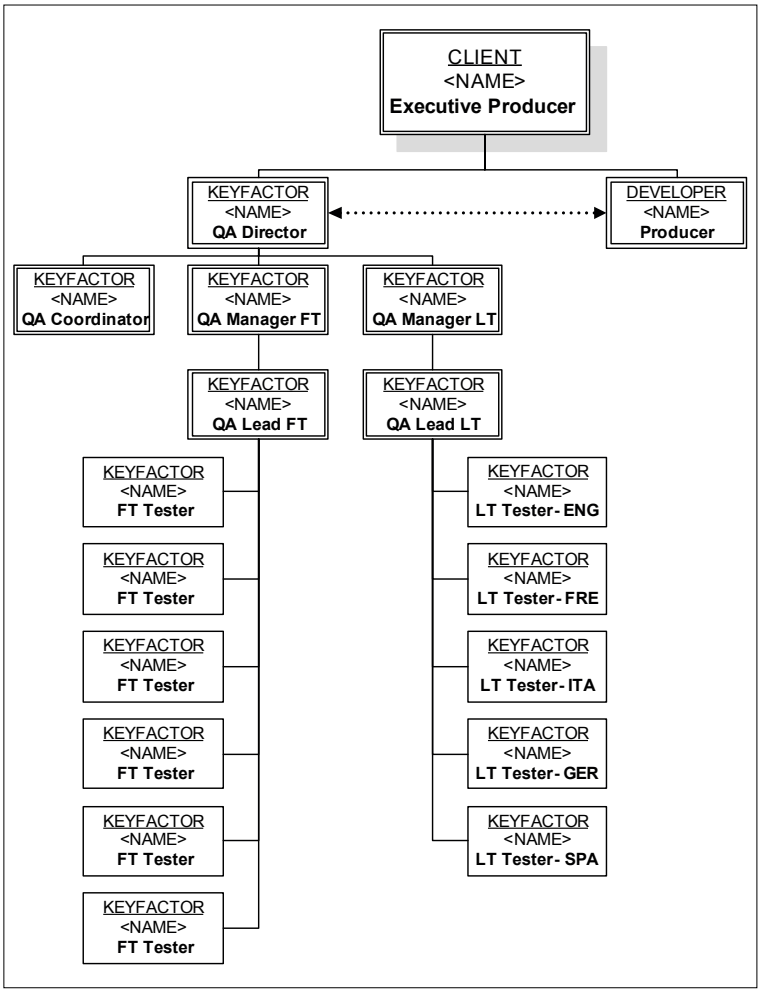


Abbildung 2: Das Organigramm einer einfachen **Project Organisation Structure**.

Role	Responsibilities
QA Coordinator	The QA Coordinator provides assistance to the test team, so that they can do their work as required. His main duties are: ● to ensure that all hardware and software meets the necessary requirements for the project. ● to provide the test team with technical support. ● to liaise with the QA Manager and the Lead Tester for their resource requirements and to manage, update and distribute the resource planning overview. ● to provide forecasts about build delivery and keep Masterlab informed. ● to provide download instructions and access data to Masterlab. ● to request FTP accounts from Masterlab when necessary.

Abbildung 3: Beispiel für eine **Rollenbeschreibung**.

Reporting

Es ist wichtig zu definieren, zu welchem Zeitpunkt welche Art von Reporting erfolgt, wer die Verantwortung dafür trägt, welche Inhalte kommuniziert werden und wer das Reporting erhalten soll (Abbildung 7).

Escalation

Da ein Projekt selten nach Plan abläuft, sollte eine Eskalationsstrategie definiert werden, die

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
16/08/2010	17/08/2010	18/08/2010	19/08/2010	20/08/2010	21/08/2010	22/08/2010
Functionality Pass #1	Functionality Pass #1	Functionality Pass #1	Functionality Pass #1	Functionality Pass #1	WEEKEND	WEEKEND
23/08/2010	24/08/2010	25/08/2010	26/08/2010	27/08/2010	28/08/2010	29/08/2010
Balancing Pass #1	Balancing Pass #1	Balancing Pass #1	Balancing Pass #1	Balancing Pass #1	WEEKEND	WEEKEND

Abbildung 4: Die zeitliche Abfolge der Tätigkeiten wird im **Test Schedule** festgehalten.

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
09/08/2010	10/08/2010	11/08/2010	12/08/2010	13/08/2010	14/08/2010	15/08/2010
				Delivery Build #1	WEEKEND	WEEKEND
16/08/2010	17/08/2010	18/08/2010	19/08/2010	20/08/2010	21/08/2010	22/08/2010
				Delivery Build #2	WEEKEND	WEEKEND
23/08/2010	24/08/2010	25/08/2010	26/08/2010	27/08/2010	28/08/2010	29/08/2010
Delivery Build #3		Delivery Build #4		Delivery Build #5	WEEKEND	WEEKEND

Abbildung 5: Die **Build-Lieferungen** werden ebenfalls in die Planung aufgenommen.

Builds	Cycle	Start	End	Test Areas	Resource Requirements
Closed Beta #1	2nd FT Pass	06/09/10	24/09/10	• Client Startup and Front End • Menus & Permanent H.U.D. • Communicator & Chat • Player Interactions • Tactics • A.I. Behaviour • Item System • Hub Progress • Regression • General Functionality Tests • Tutorial, Story Missions • Dynamic Missions New implemented Features: • Crafting System • Item Shop	5 testers/10 days = 50 man-days +5 testers/5 days (multiplayer tests) = 25 man-days
Closed Beta #2	3rd Balancing Pass	13/09/10	17/09/10	• Crafting System • Item Shop	5 testers/5 days = 25 man-days

Abbildung 6: Anhand der **Ressourcen-Aufstellung** werden später auch die Kosten abgeleitet.

Reporting Type	Method / Tool	Frequency / Schedule	Information	Responsibilities
Budget Report	Mail	Weekly	Evaluation of the costs, tracking expenses and variations from planning.	QA Manager reports to Publisher cc: QA Director
Weekly Report	Mail	Daily	Report of the week of FT. Sums up the week's testing (main areas of concern, issues covered, type of testing performed, pending feedback from developer), which areas have been covered, what is the plan for next week.	Written by Lead Tester, reviewed by QA Manager, sent to Publisher cc: QA Director, Developer
Daily Report	Mail	Weekly	Report of the day of FT. Major issues, hindering bugs and main risks will be highlighted. Tested areas, type of testing to be mentioned.	Written by Lead Tester, reviewed by QA Manager, sent to QA Director

Abbildung 7: Mithilfe des **Reporting Plans** werden Kommunikationswege und Verantwortlichkeiten festgelegt.

von den Projektbeteiligten befolgt wird.
Beispiel:

- As soon as it becomes clear that the testing will exceed the planned budget (i.e. more resources have been added for covering a specific game area, etc...), the QA Manager will contact the client via email, stating the project needs and offering solutions to overcome the issue. The budget change will then be approved (or not) by email.
- In case there is a hindering issue in the build that prevents the testing team from working on the game, the problem will be raised either via email or phone to the publisher immediately. This way, the publisher can forward the information to the developers, then possibly provide us with a solution (for instance, a fix from the developer, a tip for installing the game, etc...), or advise us how to proceed in the test (if no fix, FT team will work on the previous build etc...).
- In a generic way, problem escalations (such as a tester's incorrect behavior or mistakes) should always be communicated to the QA Manager, so they can be addressed and followed up as requested. The consequent actions will be taken by the QA Manager and the Lead Tester for the problem not to occur again.

Communication

Während eines Projekts erfolgt eine ganze Reihe von Kommunikation. Im Test Plan sollten alle offiziellen Kommunikationswege beschrieben werden. Es sollte auch der Hauptverantwortliche definiert werden und wer in die Kommunikation eingebunden wird (Abbildung 8).

Training Plan

Weil es in einem Team Erfahrungsgefälle geben kann und einige Testaufträge besondere Kenntnisse voraussetzen, ist die Beschreibung der Trainingsanforderungen wichtig (Abbildung 9).

Purchasing Plan

Wenn für ein Projekt bestimmte Anschaffungen nötig sind, sollte dies im Test Plan ebenfalls aufgeführt werden. Normale Verbrauchsgüter wie Toner, Papier usw. gehören hier allerdings nicht hinein (Abbildung 10).

Test Case

Elementare, funktionale Softwaretests werden durch Test Cases beschrieben und dienen den Testern als Anleitung, um etwa die in einer Spezifikation zugesicherten Eigenschaften zu testen. Sie sind also ein Arbeitspaket für einen oder mehrere Tester.

Der Testmanager sollte immer versuchen, über die Test Cases den gesamten Scope für Funktionstest abzudecken. Am besten eignet sich hierzu ein Produktstrukturplan (PBS/Product Breakdown Structure), in dem das Produkt in viele Einzelfunktionen heruntergebrochen wird. Dabei wird nach dem Top-down-Prinzip vorgegangen, ein größeres Feature wird in immer kleinere

Subfeatures heruntergebrochen. Als Orientierung dienen in der Regel das Spiel, die Design-Dokumentation oder User Stories. Ein Mindmapping Tool wie Mindmanager, FreeMind o.Ä. eignet sich hervorragend (Abbildung 11).

Da ein Videospiel über Hunderte von Funktionen verfügt, ist es nahezu unmöglich, das Projekt komplett zu erfassen und das gesamte Spiel abzubilden. Ein weiterer wesentlicher Aspekt ist, dass meist weit vor dem Teststart mit der Planung begonnen wird und gewisse Teile des Spiels noch nicht ausdefiniert sind. Das Herunterbrechen erfolgt erst nach und nach; je mehr man sich also mit dem Projekt vertraut macht, desto mehr erschließt sich einem das Spiel. Diese Technik wird rollierende Planung genannt.

Es kommt auch sehr häufig vor, dass nicht immer alle Features und selbstverständliche Funktionen vom Designteam dokumentiert werden. Wenn man während des Testens auf unbekannte oder nicht dokumentierte Funktionen stößt, müssen die Test Cases erweitert werden.

Bestimmte Funktionsbereiche, die nicht ausreichend dokumentiert sind, können durch »Exploratives Testen« erfasst werden. Hier erfolgen simultan Testfallentwurf, Testdurchführung, Testprotokollierung und Lernen, auf Grundlage des vereinbarten Scopes und innerhalb von festgelegten Zeitfenstern.

Wenn man alle Funktionalitäten so weit wie möglich erfasst hat, erstellt man zu jeder Funktion eine Testbeschreibung, die vorgibt, wie die entsprechende Funktion zu überprüfen ist.
Es empfiehlt sich, schon in der Entwicklung an das Testen zu denken und bei der Feature-Beschreibung auch gleich die Testbeschreibung zu dokumentieren. Am besten eignet sich für die Testfallerstellung eine Spreadsheet-Software wie etwa Excel, Numbers oder Calc. Ein Testfall

Type of Communication	Method / Tool	Frequency / Schedule	Information	Participants / Responsibilities
Kick-off Meeting	Meeting	Once at the beginning of the project	Meeting the involved stakeholders, discussion about the test planning, dev's milestones, reports and general communication.	QA Director, QA Manager, Lead Tester, Client and Developers
Problems	Mail/Phone	On occurrence	Major issue that prevents FT team from testing, a possible solution can be provided by dev.	From FT team to Lead Tester, from Lead Tester to QA Manager, QA Manager to Client
General Communication	Skype	Daily	If desired, we could use Skype to chat and be in direct touch with the main stakeholders of the project.	Developer to Lead Tester or QA Manager

Abbildung 8: Im Test Plan werden auch die **Kommunikationswege** und die Verantwortlichkeiten festgelegt.

Knowledge Area	Training type	Timing	Responsible	Trainees
TRC Compliance Testing	In-house, face to face session	2 weeks prior to project start	Compliance Manager	Not yet trained Testers
ISTQB Foundation	Seminar	Immediately	QA Manager	Non-certified Testers

Abbildung 9: In dieser Aufstellung werden die **Trainingsanforderungen** der Teammitglieder beschrieben.

Item	Unit costs	Units	Total costs
19" Flat TV – LG 19LE3300	199,00 €	2	398,00 €
Xbox 360 Kinect	149,99 €	4	599,96 €
Xbox 360 Memory Unit (64 MB)	12,90 €	8	103,20 €
DevTrack User Access License (fixed)	230,00 €	2	460,00 €
		Total Costs	**1.561,16 €**

Abbildung 10: Der **Purchasing Plan** enthält alle notwendigen Anschaffungen für das Testteam, ausgenommen sind dabei allerdings normale Verbrauchsgüter.

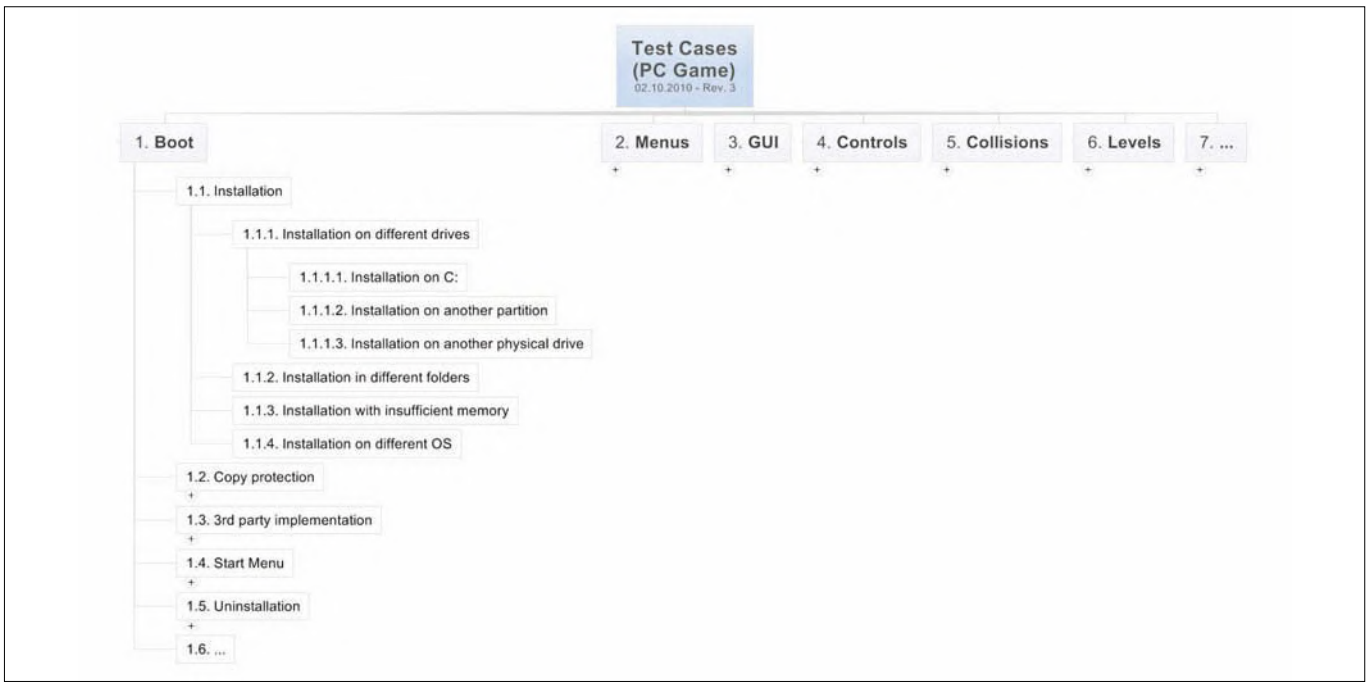


Abbildung 11: Mithilfe eines Mindmapping-Tools werden die **Test Cases** als Arbeitspakete visualisiert.

sollte folgende Informationen beinhalten:

- Test Case ID
- Test Case Summary
- Prerequisites
- Output
- Used Time
- Assigned Tester(s)
- Comments
- Category / Test Area
- Setup
- Input
- Planned Time
- Status
- Bug ID(s)

Test Case ID

Jeder Test Case sollte über eine eindeutige ID verfügen, damit er später einfacher zu identifizieren ist. Es ist unbedingt darauf zu achten, dass keine dynamische Nummerierung verwendet wird.

Beispiele:

- 01-Boot-FT-024
- 02-Menus-FT-011
- 03-GUI-FT-112
- etc.

Category / Test Area

Es empfiehlt sich, auch die Test Cases in Kategorien oder Subkategorien einzuteilen. Wenn man eine Spreadsheet-Software verwendet, kann die Hauptkategorie auf verschiedene Sheets (Tabs) aufgeteilt werden, während man dann in diesem Feld die Subkategorie angibt. Wenn man auf das obige PBS-Beispiel schaut, so wären die Hauptkategorien alle Objekte der Stufe 1; Boot, Menus und GUI beispielsweise samt den Subkategorien wären Objekte der Stufe 2 und 3. Beispiele für Subkategorien von Boot:

- Installation
- Copy Protection
- 3rd Party Implementation
- Start Menu
- Uninstallation
- etc.

Test Case Summary

Dieser Punkt beschreibt in Kurzfassung, was getestet werden soll. Wenn man wieder das erstellte PBS als Vorlage nimmt, so wären dies Objekte der Stufe 4. Beispiele für Test Case Summary für die Kategorie »Boot/Installation/Installation on different drives«:

- Installation on C:
- Installation on another partition
- Installation on another physical drive

Setup

Setup beschreibt, welche technischen Voraussetzungen erfüllt werden müssen, um diesen Test durchführen zu können. Diese Voraussetzungen können etwa bestimmte Hardware, Betriebssysteme, Software, Konfigurationen oder Ähnliches sein. Beispiel:

- PS3 Testkit with Firmware xx.xx.xx
- Console set to Debug
- Region set to PAL
- Display Language set to English
- etc.

Prerequisites

Prerequisites sind die Schritte, die der Tester erfüllen muss, bevor er den eigentlichen Testschritt ausführen kann. Es empfiehlt sich, die Schritte in der richtigen Reihenfolge und nummeriert anzugeben. Beispiel für einen Test im Option Menu:

1. Boot game
2. Press Start Button to enter the Main Menu
3. Select »Options« and press X-Button

Input

Hier wird der eigentliche Testschritt beschrieben bzw. welche Eingabeinformationen an das Programm erfolgen müssen. Auch hier sollte man die Schritte chronologisch und nummeriert angeben. Beispiel für einen Test zur Prüfung, ob die Gamma-Einstellungen für »Hell« im Optionsmenü funktionieren:

1. Press the Down-Button on D-Pad or hold down Left-Stick.
2. Hold it until the marker is down to »Gamma Correction«.
3. Press X-Button to activate Gamma setup.
4. Press Left-Button and Right-Button on D-Pad to move slider.
5. Move Left-Stick to left and hold it until reaching the left end of the indicator.
6. Then move it to the right and hold it until reaching the right end of the indicator.
7. Leave the indicator at a different position than before you started the test.
8. Press the Triangle-Button to leave the Gamma setup.
9. Press the Triangle-Button again to go back to leave the Option Menu.
10. Enter again the Option Menu.

Output

Dieser Punkt beschreibt detailliert die erwartete Funktion. Wenn das Testergebnis hiervon abweicht, handelt es sich um einen Fehler im Programm. Beispiel für einen Output bei der Prüfung der Gamma-Korrektur:

1. When holding the Left-Button on D-Pad or holding the Left-Stick left
   - **a.** the indicator moves step by step to the left.
   - **b.** the screen gets a step darker by each step left.
   - **c.** when the indicator reaches the left side it stops and the screen should not get darker.
2. When holding the Right-Button on D-Pad or holding the Left-Stick right.
   - **a.** the indicator moves step by step to the right.
   - **b.** the screen gets a step brighter by each step left.
   - **c.** when the indicator reaches the right side it stops and the screen should not get brighter.

3. When pressing the Triangle-Button with activated Gamma setup
   a. the settings will be saved.
4. When leaving the Option Menu
   a. the screen does not get darker or brighter.
5. When entering the Option Menu again
   a. the settings of the Gamma Correction have not changed.

Anmerkung: Bei der Beschreibung von Input und Output gilt es, eine dem Testteam angepasste Granulierung zu finden, um Aufwand und Nutzen im Gleichgewicht zu halten. Das obige Beispiel wäre für erfahrene Tester viel zu kleinteilig. Der Testmanager könnte zum Beispiel voraussetzen, dass der Tester weiß, mit welchen Buttons/Sticks er den Marker bewegen kann und wie Funktionen angewählt bzw. abgewählt werden. Man könnte auch davon ausgehen, dass er durch seine bisherige Berufserfahrung weiß, auf was er zu achten hat und nicht jeden Schritt vorgegeben bekommen muss. So könnte etwa eine »Input/Output«-Beschreibung für einen erfahrenen Tester aussehen:

Input
1. Select »Gamma Correction«.
2. Move indicator to left and right.
3. Select a different position than before.
4. Leave and return to Option Menu.

Output
1. The screen gets brighter and darker when moving the indicator.
2. The indicator stops when reaching the edges.
3. The indicator's position and game settings are saved when leaving and returning to the Menu.

Bei der Erstellung von Test Cases spielt also der Erfahrungsschatz des Testteams eine wesentliche Rolle. Der Testmanager muss die Fähigkeiten des Teams entsprechend einschätzen können.

Planned Time / Used Time

In diesem Abschnitt wird angegeben, wie lange die Bearbeitung eines Test Case dauern darf/könnte/sollte und wie lange der Tester tatsächlich benötigt hat (Abbildung 12). Die Verwendung von Zeiten auf Testfallebene hat dabei sowohl Vor- als auch Nachteile.

Vorteile	Nachteile
● Bessere zeitliche Einschätzung bei Planung und Kosten ● Zusätzliche Performance-Informationen bei der Bewertung von Teams und Testern ● Zusätzliche Tracking-Informationen für die Überwachung der aktuellen Coverage	● Die Zeiten können Fehler beinhalten. ● Die Zeiten können durch Abstürze, Fehler usw. in der Testbuild nicht einzuhalten sein. ● Die Erfassung der Zeiten bedeutet zusätzlichen Aufwand und zusätzliche Software.

Abbildung 12: In diesem Abschnitt werden die **geplanten und tatsächlichen Bearbeitungszeiten** eines Test Case festgehalten, das hat sowohl Vor- als auch Nachteile.

Bei den meisten Projekten verzichten wir auf die Erfassung von Zeiten auf Testfallebene. Stattdessen erfassen wir Zeiten auf höheren Ebenen. Bezogen auf das PBS wäre dies auf den Ebenen 1 und 2.

Status

Um den aktuellen Stand der Bearbeitung zu kennen, sollten Test Cases mit einem Status versehen werden. Auf dieser Grundlage kann der Testmanager sein Reporting aufbauen. Abhängig von der Granulierung des Reportings sollten die Statusauswahlmöglichkeiten mit Bedacht gewählt werden. Für manche Projekte/Unternehmen könnte es durchaus reichen, nur einen »Fail/Pass«-Status zu verwenden. Um an dieser Stelle keine Fehler zu generieren, sollte dies ausschließlich über ein Drop-down-Menü erfolgen. Beispiele für Status:
- not started yet
- in progress
- bad test case
- not doable
- passed
- failed

Assigned Tester(s)

In diesem Feld gibt der Testmanager an, welcher Tester den Testfall bearbeiten soll bzw. bearbeitet hat. Damit kann man bei Rückfragen den entsprechenden Tester ansprechen, bei einem erneuten Durchlauf dieselbe Person an dem Testfall arbeiten oder einen anderen Tester diesen durchführen lassen. Auch hier gilt, niemals Namenskürzel zu verwenden. In der Praxis hat sich zudem gezeigt, dass auch hier ein Drop-down-Menü von Vorteil ist, um etwa später Performance Reviews auf Testerebene durchführen zu können.

Bug IDs

Wenn bei der Bearbeitung von Test Cases Fehler gefunden werden, dann trägt der Tester die entsprechenden Bug IDs oder Ticketnummern hier ein. Dies erlaubt eine eindeutige Zuordnung.

Comments

Dieses Feld ist für den Tester gedacht, um Kommentare für den Testmanager zu hinterlassen. Die Kommentare können Feedbacks zum Test Case selbst oder zusätzliche Informationen sein. Es könnten auch qualitative Informationen zu den Funktionen hinterlassen werden. Beispiele:
- Option Menu is not accessible
- Need more info in the Input Description
- The difference between dark and bright is not very huge

Fazit

In diesem ersten Artikel wurden die Abhängigkeiten zwischen der Dokumentation auf Unternehmens- und Projektebene erklärt. Zudem wurden mit Test Plan und Test Case die beiden wichtigsten Dokumente für die Planung besprochen. Alle weiteren Dokumente bauen auf diese beiden auf und werden in den nächsten beiden Teilen der Reihe besprochen. Im letzten Artikel werden schließlich alle Dokumente in einen zeitlichen Kontext gebracht.

Frank Fay

»Wir sind wie Opel und General Motors«

Im Gespräch mit Zynga Germany

Eben noch ein kleines Mainzer Startup und plötzlich ein Teil von Zynga. Die Gründer der Dextrose AG erzählen von gescheiterten Investorengesprächen, wie es zur Übernahme durch den Social-Games-Giganten kam und welche Pläne Zynga mit seinem frisch erworbenen deutschen Studio hat.

Rocco Di Leo

ist CEO von Zynga Germany.

Paul Bakaus

ist CTO von Zynga Germany.

WWW
makinggames.de
- Finanzierung eines Startups

> Wir haben zu Anfang nicht gedacht, dass wir unser Konzept mehrfach umschmeißen würden.

Making Games Die Dextrose AG wurde ja erst vor rund einem Jahr gegründet. Wie habt ihr es geschafft, dass ihr euch schon jetzt in Zynga Germany umtaufen könnt? War das von Anfang an euer Plan?

Rocco Di Leo Die Dextrose AG hatte ursprünglich einen ganz anderen Hintergrund. Ich wollte eigene Projekte machen, vor allem Webprojekte in Kooperation mit anderen Partnern und weniger Games. Ich hatte die Dextrose AG zunächst allein gegründet, habe aber kurz danach Paul wiedergefunden – ein alter Kollege von mir, mit dem ich schon vor ein paar Jahren zusammengearbeitet hatte. Schon damals hatte er die Idee von einem an Zelda und Final Fantasy angelehnten Spiel, das im Browser ohne Plugin-Download stattfinden sollte. Er hatte auch schon kleinere Proof of Concepts gebastelt und in der Zwischenzeit das Konzept in Richtung Social Community weiterentwickelt.

Als wir uns dann Ende 2009 zufällig bei einer Veranstaltung trafen, bot ich ihm an, doch einfach gemeinsam in dieser Richtung etwas zu machen und eine grafische Social Community zu entwickeln – quasi ein Facebook als isometrisches Game ohne Plugins oder Client Download. Paul kam also schon früh an Bord, daher sehen wir uns auch definitiv als Gründerteam. Und dann ging es los mit unserer Mission: Wir mussten in kürzester Zeit einen richtig guten Prototyp basteln und ihn wichtigen Leuten, am besten der ganzen Welt demonstrieren, um am Ende die notwendige Finanzierung für unsere »Aves Community« zu erhalten. Wir haben zu diesem Zeitpunkt natürlich im Traum nicht daran gedacht, dass wir das Konzept mehrfach komplett umschmeißen würden.

Making Games Welche Lücke sollte denn die Aves Engine auf dem Browsergames-Markt schließen, was war eure Vision?

Paul Bakaus Im Grunde wollten wir es ermöglichen, grafisch schöne Spiele zu entwickeln, die keinen Client Download oder spezielle Plugins wie Flash oder Unity benötigen und problemlos auf mobilen Plattformen laufen. Das war mit den Mitteln, die wir Ende 2009 sahen, einfach nicht möglich. Diese Lücke wollten wir schnell schließen, vor allem weil wir auch von Entwicklerseite immer wieder gesagt bekamen, dass ein Bedarf nach solch einer Technologie durchaus vorhanden war und immer noch ist. Die Firmen wollen expandieren, sie wollen neue Märkte erschließen und das heißt, dem Spieler so wenige Einstiegsbarrieren wie möglich in den Weg zu legen. Für uns bedeutet dies: URL eintippen, egal auf welchem Endgerät, und sofort ein grafisch schönes Spiel starten können. Wir wollten also auch ohne Browser Plugins mindestens die Qualität moderner Flash Games erreichen.

Making Games Ist euer Ansatz mit den offenen Standards wie HTML5 und JavaScript auch Teil der Zynga-Linie? Werdet ihr genau so weitermachen oder müsst ihr euer Konzept umstellen?

Paul Bakaus Wir machen im Prinzip genau so weiter wie bisher. Wir entwickeln weiterhin technische Bausteine, man könnte uns also als »Research & Development«-Studio bezeichnen.

Rocco Di Leo Richtig, im Moment entwickeln wir keine Spiele für Zynga, wir bauen Komponenten. Das ist im Prinzip die Game Engine, die wir sowieso immer schon entwickeln wollten. Wir sind im Unternehmen eine von vielen Initiativen, die vielversprechende Technologien evalu-

Wir haben Rocco Di Leo (rechts) und Paul Bakaus (links) auf dem **Browsergames Forum** in Offenbach getroffen und mit ihnen über ihr turbulentes letztes Jahr und die Zukunft von Zynga Germany gesprochen.

ieren. Natürlich wird auch bei Zynga in alle Richtungen geforscht und ausprobiert. Am Ende kann es daher sein, dass unsere Game Engine in einem der nächsten Projekte eingesetzt wird oder aber auch erst einmal eine andere Technologie zum Einsatz kommt.

Making Games Ist damit dann ausgeschlossen, dass die ehemalige Aves Engine als Lizenzprodukt erscheint?
Rocco Di Leo Diese Option ist erst einmal vom Tisch, richtig. Allerdings wissen nur wenige, dass Zynga selbst auch Open-Source-Projekte entwickelt oder sich an ihnen beteiligt.
Paul Bakaus Daher ist es auch eher realistisch, dass irgendwann Teile unserer Game Engine Open Source werden könnten, als dass sie an Dritte kommerziell lizensiert wird.

Ich selbst arbeite schon seit Jahren an Open-Source-Projekten, und dieses Thema ist mir natürlich auch persönlich sehr wichtig. Die Entwicklung einer »exklusiven« Zynga-Technologie sieht da natürlich erst einmal wie eine Kehrtwendung aus. Ich würde mich aber freuen, wenn wir irgendwann Teile als Open Source freigeben könnten. Konkrete Pläne dazu gibt es aber noch nicht.

Making Games Was sind denn eurer Meinung nach die Gründe, warum es bislang keine ausgereiften Middlewares für den Browser gibt, wie sie etwa im Konsolenbereich üblich sind?
Paul Bakaus Die Frage haben wir uns natürlich auch gestellt. Ein Grund ist, dass die Browsertechnologie dafür lange Zeit noch nicht reif genug war. Die ursprüngliche Vision, die ich damals an Rocco herangetragen hatte, war allerdings nicht neu. Ich hatte sie schon einige Zeit weiterentwickelt und jedes Jahr versucht, einen Prototyp zu programmieren. Und erst letztes Jahr haben wir dann entschieden, dass die Zeit für unser Produkt gekommen war. Die Browser sind durch neue JavaScript-Engines und Performance-Upgrades so schnell geworden, dass es erstmals wirklich Sinn ergab, unsere Idee in die Tat umzusetzen. Der zweite Punkt ist, dass in der Regel Web-Entwickler keine guten Spiele-Entwickler sind – und das Gleiche gilt auch umgekehrt. Es gibt also einfach sehr wenige Talente, die überhaupt in der Lage sind, ein solches Projekt im Browsergame-Bereich zu bauen.

Wir dachten auch, dass selbst ohne Middleware-Markt viele große Firmen bestimmt schon interne Lösungen entwickelt hätten, dem war aber offenbar in vielen Fällen nicht so. Das hat uns natürlich zusätzlich motiviert, unsere Idee zügig weiterzuverfolgen.

> Wir wollten auch ohne Browser Plugins die Qualität moderner Flash Games erreichen.

Making Games Ihr hattet also die Vision und wusstet auch, dass es Bedarf in der Branche für eure Technologie gab. Wie seid ihr dann mit eurem Produkt auf potenzielle Investoren und Publisher zugegangen?
Rocco Di Leo Wir wollten ja ursprünglich die Aves Community auf Basis unserer Technologie entwickeln. Auch wenn wir sehr viele tolle Ideen für die Community hatten, war uns schnell klar, dass ein solches Projekt, das technisch ambitioniert war und dem kein ordentliches Geschäftsmodell zugrunde lag, bei Investoren und Venture Capitalists nicht besonders gut ankommen würde. Weil wir aber auf jeden Fall Geld und mehr Leute brauchten, mussten wir unsere Strategie ändern. Wir haben uns zunächst die wenigen Plugin-freien Spiele der großen Browsergame-Studios angeschaut und kamen zu dem Schluss, dass eine Game Engine, die das leistet, was wir in der Aves Community abbilden wollten, einen bis dato nicht existenten Middleware-Markt erschließen könnte. Niemand hatte etwas in der Hand, nicht die Browsergame-Hersteller und auch nicht die großen Internetfirmen wie Google oder Yahoo.

Zuträglich war uns dabei auch der HTML5-Hype, der zu diesem Zeitpunkt nur am Rande etwas mit unserer Technologie zu tun hatte. Wir

Derzeit setzen Zyngas Spiele wie **Farmville** vor allem auf Flash, das könnte sich mit dem Einsatz offener Standards aber in Zukunft ändern: »Natürlich wird auch bei Zynga in alle Richtungen geforscht und ausprobiert. Am Ende kann es sein, dass unsere Game Engine in einem der nächsten Projekte eingesetzt wird oder aber auch erst einmal eine andere Technologie Verwendung findet.« (Rocco Di Leo)

> Web-Entwickler sind in der Regel keine guten Spiele-Entwickler – und umgekehrt.

Zynga Germany

Rocco Di Leo hat Ende 2009 die Dextrose AG gegründet, Paul Bakaus kam wenig später als CTO an Bord. Kernprodukt war die Aves Engine, die erste professionelle Middleware zur Erstellung von Browsergames ohne Plugins. Die Aves Engine nutzte den kompletten Open Web Stack und verzichtete bewusst auf Erweiterungen wie Flash oder Unity. Dadurch sollten nicht nur Einstiegsbarrieren beim User eliminiert, sondern auch eine echte Cross-Platform-Distribution auf Desktop und mobilen Endgeräten ermöglicht werden.

Das zweite Projekt der Dextrose AG war die Aves Community, eine soziale Plattform wie Facebook, die komplett in einer isometrischen Welt abgebildet wurde. Mit der Übernahme durch Zynga wurden nun beide Projekte in ihrer ursprünglichen Form eingestellt, eine Lizenzlösung existiert daher nicht mehr. Zynga Germany entwickelt die Technologie zwar weiter, fungiert aber mittlerweile als interner Dienstleister im Browser- und Social-Games-Bereich.

hatten also auch das Glück der Tüchtigen, waren zum richtigen Zeitpunkt mit einem funktionierenden Prototyp im Gespräch und wurden mit Anfragen aus aller Welt überhäuft.

Jetzt mussten wir die ganze Vision der Aves Engine »nur« noch finanzieren, also haben wir Venture-Capital-Firmen kontaktiert. Mit Präsentationen und Businessplänen im Gepäck besuchten wir VCs im In- und Ausland. Hier wurde sinngemäß immer die gleiche These aufgestellt: »Das sieht ja gut aus, aber das machen doch sicher Google oder Zynga mal eben innerhalb von drei Wochen mit ihren Tausenden Mitarbeitern.« Da half es auch nicht, dass wir schon konkret Letter of Intents von potenziellen Lizennehmern vorweisen konnten, die ein fertiges Produkt einsetzen wollten. Hier war also kein Vorankommen. Bei den Spielestudios sah es da schon anders aus. Sie wussten, dass sie intern nicht leisten konnten, was wir ihnen anboten, scheuten sich aber davor, in uns zu investieren. Da hieß es dann also: »Klingt super! Kommt wieder, wenn ihr fertig seid und wir machen was zusammen.« Das war also unser Dilemma: Der Prototyp kam super an, wir waren dazu bereit, das richtige Produkt zu bauen, der Markt war da, aber wir hatten nicht genügend finanzielle Unterstützung. Wir mussten erneut umdenken. Eine hinausgeschobene Option für uns war dann die Übernahme der Dextrose AG durch ein anderes Unternehmen, was das Aus für das geplante Lizenzierungsmodell bedeutete.

Making Games Und wie genau liefen eure Gespräche mit Investoren ab? Welche Faktoren spielten bei den Verhandlungen eine Rolle?
Rocco Di Leo Der Türöffner war ohne Frage unser Prototyp. Paul hatte den ersten schon Ende April 2010 auf der jQuery-Konferenz vorgestellt. Den zweiten Prototyp haben wir dann auf der E3 präsentiert. Daraufhin kamen viele Firmen auf uns zu, unter anderem auch Zynga, die uns mal kennenlernen wollten. Wir sind dann nach San Francisco geflogen und hatten zu diesem Zeitpunkt eigentlich nur im Sinn, Zynga als Lizenznehmer für die Aves Engine zu gewinnen. Es hat sich dann aber überraschenderweise herausgestellt, dass Zynga tatsächlich an einer kompletten Übernahme interessiert war und weniger an einem einzelnen Produkt. Wie im Silicon Valley üblich, versucht man durch Übernahmen nicht nur ein Produkt, sondern vor allem auch gleich die Talente dahinter zu akquirieren.

Making Games Ihr habt also direkt zugesagt?
Rocco Di Leo Nein, wir waren ehrlich gesagt am Anfang doch etwas überrascht über das Angebot. Zu diesem Zeitpunkt war die Option einer Übernahme noch nicht wirklich in unseren Köpfen präsent. Wir glaubten so fest an unser Lizenzierungsmodell, dass wir nie ernsthaft über eine andere Option gesprochen hatten. Wir waren daher auch ziemlich unsicher, wie man so einem Vorschlag entgegentreten sollte. Die Chance, für die weltweit größte Social-Game-Firma zu arbeiten, war unglaublich verlockend. Hinzu kam, dass die Leute, die wir kennenlernten durften, alle super drauf und sehr smart waren. Die gesamte Firmenkultur entsprach insgesamt einem Ideal, das wir selbst immer für unsere wachsende Firma im Sinn hatten.

Wir mussten uns also erst einmal selbst darüber klar werden, was eine Übernahme für unsere gesamte bisherige Planung bedeutete und baten Zynga daher um Bedenkzeit. Wir wollten eigentlich ein Unternehmen aufbauen und mit vielen Firmen zusammenarbeiten, wir fürchteten eine Abhängigkeit. Zurück in Deutschland überlegten wir also nochmals, was wir eigentlich für persönliche Ziele hatten.
Paul Bakaus Im Grunde mussten wir noch einmal in uns gehen und uns Gedanken machen, was genau eine Übernahme für uns bedeuten würde. Es war klar, dass es riesige Vorteile haben würde, eine Firma mit mehr als enormen Talenten und Ressourcen hinter sich zu haben. Uns hat eingeleuchtet, dass wir die Chance auf eine enorme Beschleunigung der Entwicklung haben würden. Wir hatten die Möglichkeit, auf einmal viel schneller zu unserem eigentlichen Ziel zu gelangen, nämlich Millionen Menschen mit unserer Technologie zu erreichen.
Rocco Di Leo Ich denke, genau dieser Punkt war uns vorher nicht klar, als uns das Angebot gemacht wurde. Vielleicht waren wir ein bisschen zu egoistisch und haben gedacht, wir würden das alles auch selbst schaffen, jeder findet unseren Prototyp schließlich super und der unternehmerische Erfolg käme praktisch von selbst. Vielleicht hätte das auch gestimmt, aber es hätte auch unglaublich lang gedauert. Bei Zynga sind wir jetzt in einem Unternehmen, das es versteht, schnell am Markt zu agieren, Innovation zu evaluieren, zu fördern und praktisch unli-

mitierte Ressourcen für das bestmögliche Ergebnis in kürzester Zeit bereitzustellen. Das ist also genau das, was wir für den Aufbau unserer Technologie brauchen.

Im Nachhinein fragt man sich natürlich, warum wir überhaupt so skeptisch waren. Ich denke, wir hatten einfach Angst, unsere Unabhängigkeit zu verlieren. In vielen Gesprächen mit den Leuten in San Francisco hat sich aber schnell herausgestellt, dass die gesamte Firmenkultur darauf aufbaut, die eigenen Leute selbstständig agieren zu lassen. Bei Zynga ist fast jeder in irgendeiner Form ein Entrepreneur, und dieser Stil hat uns sehr zugesagt.

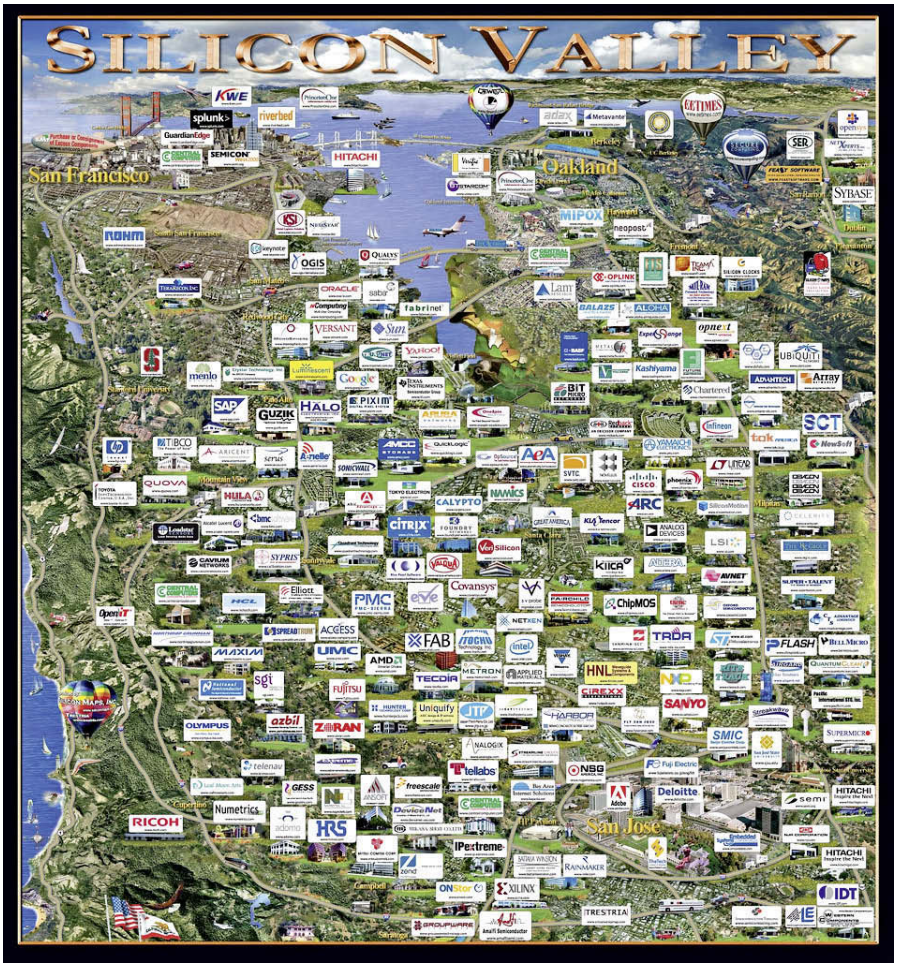
Ein großer Wunsch war es auch, dass wir ein eigenes Studio innerhalb von Zynga bilden, sprich Zynga Germany als erstes europäisches Entwicklerstudio überhaupt im Unternehmen aufbauen können. Wir wollten uns auf Research & Development konzentrieren, und auch das war in Zyngas Interesse. Operativ hat sich für uns also nicht viel geändert. Wir erforschen und entwickeln weiterhin Technologien, vielleicht irgendwann auch mal Spiele.

Making Games Zynga schließt also nicht von vornherein aus, dass bald auch in Deutschland Spiele für das Unternehmen entstehen?
Rocco Di Leo Sicherlich nicht. Das ist zwar heute noch nicht auf dem Radar, aber spätestens wenn wir die ersten Grafiker-Stellenausschreibungen veröffentlichen, ist das ein untrügerisches Zeichen dafür, dass wir loslegen. (lacht)

Im Ernst: Ich denke, es liegt auf der Hand, dass wir als Entwickler der Komponenten, die in Spielen anderer Studios innerhalb von Zynga zum Einsatz kommen könnten, auch selbst irgendwann einmal eigene Games entwickeln möchten. Für dieses oder nächstes Quartal steht das aber definitiv nicht auf dem Plan.

Making Games Und was steht dann für dieses und das nächstes Quartal auf dem Plan? Wie sieht eure Rolle in der Zynga-Strategie aus?
Rocco Di Leo Wir haben die Aves Engine immer als Cross-Platform-Lösung beworben, die ohne Plugins auf allen modernen Endgeräten funktioniert. Jetzt und im kommenden Quartal werden wir weitere Mitarbeiter einstellen und die ersten Komponenten für einen möglichen Produktiveinsatz fertig entwickeln. Wir wollen in Deutschland kein riesiges Team aufbauen, sondern vielmehr eine kleine »Research & Development«-Einheit bleiben und praktisch mit den Aufgaben wachsen.

Making Games In Bezug auf die Aves Engine habt ihr mal erwähnt, dass ihr damit »sehr große, epische Single- und Multiplayer-Games« ermöglichen wollt. Werden wir von Zynga also demnächst neben kleineren Titeln auch hochkomplexe Blockbuster als Browsergames sehen?
Paul Bakaus Wie bereits gesagt: Bei Zynga gibt es sehr viele Initiativen, es wird sehr viel geforscht. Wir werden daher adaptiv vorgehen und in erster Linie schlicht exakt die Technologie entwickeln, die Zynga braucht.



»Den Deutschen fällt es extrem schwer, eine ähnliche Kultur wie in Silicon Valley aufzubauen. Dort hast du Anwälte, die auf Startups spezialisiert sind, dort sitzen wahnsinnig viele VCs und natürlich auch Gründer, die später selbst zu Investoren werden – das ist ein hervorragendes Netzwerk.« (Paul Bakaus)

Rocco Di Leo Richtig, Zynga besteht aus sehr vielen Studios, und dort wird mit vielen verschiedenen Technologien gearbeitet, mit denen wir derzeit wiederum nichts zu tun haben. Aber wenn es um Plugin-freie Browsertechnik geht, egal ob für 2D- oder 3D-Titel, dann werden wir entsprechende Komponenten liefern, wenn wir die dafür notwendige Reife sehen.

Making Games Das Ziel ist also, ganz neue Spielkonzepte mit eurer Technologie anzuschieben?
Paul Bakaus Wir versuchen zwei Dinge: Zum einen wollen wir natürlich neue Plattformen und Zielgruppen erschließen, speziell im Smartphone-Segment. Zum anderen werden wir hoffentlich bald Features entwickeln, die Entwickler inspirieren, neue Konzepte umzusetzen.
Rocco Di Leo Ein klassisches Beispiel dafür wäre die Physik. Wir entwickeln die Komponente, und die Spielestudios können sich dann Gedanken machen, wie sie die neuen Möglichkeiten am besten einsetzen. Paul und ich sind mit Computer- und Videospielen groß geworden und wir würden gern einige Dinge, die man aus Core-Spielen kennt, sinnvoll in Casual-Spiele übertragen.

> Zynga war gleich an einer kompletten Übernahme interessiert und weniger an einem einzelnen Produkt.

> Ich bin praktisch persönlich all-in gegangen.

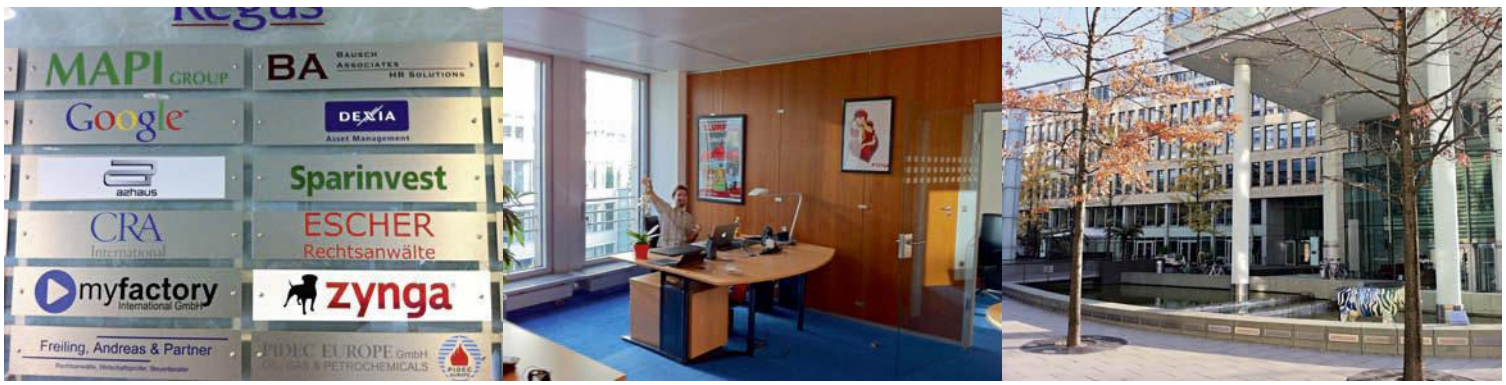
Making Games Zynga-Spiele laufen derzeit vor allem auf Facebook. Unterstützt die Plattform denn überhaupt solch eine moderne Engine mit anspruchsvoller Grafik inklusive aller Standards, oder liegt die Zukunft der Zynga-Spiele vielleicht ganz woanders als auf Facebook?
Paul Bakaus Das Technische ist hier eigentlich eher nachrangig und kann im Zweifelsfall mit dem jeweiligen Partner gemeinsam gelöst werden. Zynga will für alle großen Plattformen Spiele optimal aufbereiten. Jedes Social Network hat seine Eigenarten mit oft sehr unterschiedlichem Nutzerverhalten. Diese muss man zunächst einmal verstehen, bevor man auf Facebook & Co erfolgreich sein kann. Natürlich wird weltweit gerade auf Facebook sehr viel Zeit verbracht, daher sind sie auch ein so wichtiger Partner für uns und wir arbeiten sehr eng zusammen, um ein funktionierendes Ökosystem für Spiele zu entwickeln.

Making Games Kommen wir noch mal zurück zu euren Investorengesprächen. Vorhin hattet ihr erwähnt, dass diese nicht immer einfach waren, dass es zu Missverständnissen kam und es schwierig war, das eigene Produkt zu verkaufen. Habt ihr vielleicht ein paar goldene Regeln für Gespräche mit VCs und Publishern?
Paul Bakaus Mein Tipp wäre tatsächlich, nicht auf europäische Investoren zuzugehen. Wir haben die Erfahrung gemacht, dass man dort nur äußerst schwer mit Innovation punkten kann. Es wird immer ein »Proven Business«-Modell erwartet, was natürlich nichts anderes als ein Copycat bedeutet.
Rocco Di Leo Ich kann noch als Tipp mitgeben, dass alles mit dem Prototyp steht und fällt. Bei den Browsergames ist es wohl häufig noch so, dass Entwickler einfach mit Powerpoint-Präsentationen kommen und dann stehen sie mit 100 anderen Konzepten auf dem Flur. Bei den lokalen VCs würde ich es nicht ganz so hart wie Paul formulieren. Aber Fakt ist, dass es amerikanische Startups im Valley deutlich einfacher haben, an Venture Capital für innovative Projekte zu kommen. In Deutschland wird Venture Capital ja immer noch mit »Risikokapital« übersetzt. Da erkennt man schon direkt die unterschiedlichen Herangehensweisen.

Making Games Hattet ihr im letzten Jahr, als ihr auf der Investorensuche wart, Unterstützung von deutschen Förderinstitutionen?
Rocco Di Leo Nein, das gesamte Investment habe ich persönlich aufgebracht – und davon vor allem unsere Gehälter, Reisen, Equipment, Events, aber auch Grafiker für unsere 3D-Modelle in den Prototyp bezahlt. Man muss dazu erwähnen, dass Paul und ich schon lange keine Studenten mehr sind. Wir haben auch beide Ehefrauen und bei mir kam die Geburt meines Sohnes zum Jahreswechsel hinzu. Das klassische Startup-Bootstrapping war bei uns also nicht möglich. Aber ich habe an die Vision geglaubt. Und ich habe natürlich auch an Paul geglaubt, dass er das Ganze im Frontend so hinbekommt, wie wir uns das vorstellten. Und Paul wiederum vertraute darauf, dass ich die richtigen Entscheidungen treffe, auf meinen Input in die Technik und dass von der operativen Seite alles professionell abläuft. Ich glaube, wenn dieses Vertrauen in einem Startup nicht da ist, dann kann man irgendwann keine Nacht mehr ruhig schlafen.

Ich bin praktisch persönlich all-in gegangen, was ich auch nicht jedem empfehlen würde. Aus Zeitgründen haben wir uns nicht an die Gründerfonds und Banken gewandt, wir wollten einfach sehr schnell zu einem Ergebnis kommen und nicht erst bei Dutzenden Leuten vorsprechen und durch monatelange Bewerbungsprozesse gehen. Grundsätzlich würde ich aber jedem empfehlen, sich das mal anzusehen, wenn man die Zeit aufbringen kann.

Making Games Wenn man mal zusammenfasst, dass es in Deutschland schwierig mit VCs ist, dass man als Entwickler nur umständlich an eine Finanzierung gelangt, würdet ihr sagen, dass diese Zurückhaltung den hiesigen Markt für die Zukunft kaputt macht?
Paul Bakaus Ich würde nicht sagen »kaputt macht«, das ist der falsche Ausdruck. Ich denke, er war noch nie wirklich heil. Den Deutschen fällt es extrem schwer, eine ähnliche Kultur wie in Silicon Valley aufzubauen. Dort hast du Anwälte, die auf Startups spezialisiert sind, dort sitzen wahnsinnig viele VCs und natürlich auch Gründer, die später selbst zu Investoren werden – das ist ein hervorragendes Netzwerk.



Ganz frisch hängt das Firmenschild am Eingang des **neuen Büros**. Dieser Erfolg war aber nicht von vornherein abzusehen: »Ich bin praktisch all-in gegangen«, erzählt Rocco.

Making Games Wenn ihr sagt, dass auch die Innovationen nicht richtig erkannt werden: Liegt das dann vielleicht auch an den Konzepten der deutschen Gründer, muss man hierzulande vielleicht auch inhaltlich umdenken?
Paul Bakaus Ich bin mir nicht sicher, das ist ein zweischneidiges Schwert. Ich denke, dass es sehr viele Startups mit innovativen Ideen in Deutschland gibt, die es auch bei den Investoren versuchen und dann scheitern. Auf der anderen Seite gibt es viele Gründer, die mit Copycats zu einem erfolgreichen Abschluss kommen – das bestraft dann wieder die Innovativen. Am Ende ist es so, dass du nur wenig Innovation in einem Land haben kannst, wenn man den oft jungen Leuten nicht unter die Arme greift und sie nicht bei ihren Projekten unterstützt.
Rocco Di Leo Die Alternative im Spielebereich ist natürlich, einen Publishing Deal mit irgendeinem etablierten Studio einzugehen. Das Problem ist hierbei nur, dass man in der Regel nahezu fertige Produkte sehen will und nicht selten sehr viel Einfluss auf das eigene Produkt verliert.

Making Games Bald ist Jahreswechsel, Zeit für Prognosen: Wo seht ihr Zynga Germany 2011?
Rocco Di Leo Ich persönlich würde gern sehen, dass viele Millionen Spieler unsere Technologie bewusst oder unbewusst nutzen, sei es im Browser oder auf ihren Smartphones. Außerdem würden wir als Studio gern irgendwann unser erstes eigenes Spiel produzieren.

Paul Bakaus Unsere Hauptaufgabe wird aber natürlich die Entwicklung von Technologie bleiben. Wir wollen uns als zuverlässigen »Dienstleister« innerhalb von Zynga positionieren und den einzelnen Studios das Bestmögliche an Technologie liefern, das heißt, wir haben auch eine große Verantwortung im Bereich Support und Developer Relations.
Rocco Di Leo Unsere Position sehe ich da ein wenig wie das Verhältnis zwischen der deutschen Opel AG und der amerikanischen Besitzerin General Motors. Opel betreibt viel Research & Development für GM, baut aber auch eigene Autos. Eine ähnliche Ausrichtung streben wir auch an.

Making Games Ein interessanter Vergleich, wir sind gespannt!
Vielen Dank euch beiden für das Gespräch!

Interview: Yassin Chakhchoukh

» Der deutsche Markt war noch nie wirklich heil. «

cerasus.media aus Berlin entwickelt seit 9 Jahren Casual Games und Entertainment-Software für PC, Mac, Konsolen und Smartphones. Chartplatzierungen in der ganzen Welt unterstreichen unseren internationalen Anspruch.
cerasus.media
Wir suchen
Game Designer (m/w)
Producer (m/w)
Graphic User Interface Designer (m/w)
Grafiker (m/w)
www.cerasus.de/jobs
JETZT BEWERBEN!

Den Kuchen aufteilen
Revenue Shares bei einem Flash Game

Andy Moore beschreibt im Post Mortem, wie es bei seinem Flash Game SteamBirds zu einem passenden Deal mit einem geeigneten Geldgeber kam und wie viel vom Kuchen am Ende für ihn selbst übrig blieb.

Andy Moore

ist Game Designer und Programmierer.

Wenn Andy nicht gerade mysteriös dreinschaut (siehe Foto), trinkt er nach eigener Aussage ab und an zu viel und entwickelt ansonsten Computerspiele. Sein Weg in die Industrie führte ihn über das Community Management und die Beratung diverser Unternehmen. Er liebt es, Unterricht zu geben, als Sprecher aufzutreten und generell Teil der Indie Game Community zu sein. Seiner wirklichen Leidenschaft, dem Programmieren, folgt er mit seinen eigenen Projekten, zu denen auch das Action-Strategiespiel SteamBirds zählt.
www.andymoore.ca

WWW
makinggames.de
- Erlösmodelle von Browsergames

Im März 2010 ist mein kostenloses Flash Game »SteamBirds« an den Start gegangen. Ich möchte an dieser Stelle gern einige Zahlen zum Aufbau der Kernerlöse des Spiels nennen. Dadurch soll auch deutlicher werden, nach welchen Grundregeln das Business im Universum der Singleplayer Flash Games funktioniert.

Wichtig ist noch zu erwähnen, dass einige Sponsoren nicht namentlich im Zusammenhang mit ihrem Preisschild genannt werden möchten. Ich habe also ein klein wenig die Dimensionen der Zahlenwerte verschoben und einige Angaben außen vor gelassen, sodass ich niemandem auf den Schlips trete.

Das Share-Modell

Ich habe mich ganz schön schwergetan, ein geeignetes Share-Modell zu finden. Ich wünschte, es gäbe nicht 100, sondern 150 Prozentpunkte, damit ich jedem ein bisschen mehr zahlen könnte. Aber ich denke auch, dass die gewählte Ratio einigermaßen fair für jeden ist. Doch ich muss auch ehrlicherweise zugeben, dass ich lieber etwas mehr Geld für mich selbst herausgeschlagen hätte. Wer würde das nicht wollen? Aber selbst, wenn ich jemandem etwas weniger gegeben hätte, hätte ich es doch wieder an jemand anderes verteilt und nicht für mich selbst behalten. Auf jeden Fall bin ich jetzt mit dem Endergebnis zufrieden.

Fangen wir mit dem Share der iPhone-Portierung von SteamBirds an, deren Rechte bei Semi Secret Software liegen (u.a. auch Canabalt und Gravity Hook HD):

- 30% aller Erlöse gehen zunächst an Apple.
- 50% der verbleibenden Erlöse gehen an Semi Secret Software.
- 50% gehen schließlich an das SteamBirds-Team (siehe unten).

In **SteamBirds** müssen die Spieler alle gegnerischen Flugzeuge durch geschickte Manöver ausschalten. Das ursprüngliche Flash Game ist mittlerweile auch für iPhone und iPad erschienen.

Außerhalb des iPhones gehen 10% aller Gewinne, die durch »Flash Game License« entstehen, direkt an sie zurück. Sie werden zu allererst bezahlt, denn ohne ihren exzellenten Service für den Vertrieb auf Flash-Portalen hätte ich nur einen Bruchteil des Geldes gemacht. Nicht auf sie zurückzugreifen, wäre töricht gewesen; sie nicht zu bezahlen, unverschämt.

Von dem verbleibenden Profit für das SteamBirds-Team gehen

- 50% an mich selbst, für das Game Design und die Programmierung. Ich gehe so weit zu sagen, dass ich ein paar wirkliche Kick-Ass-Lines an Code eingebaut habe. Und ich wünschte, ich könnte sie so einfach wie ein Kunstwerk herumzeigen.
- xx% gehen an Daniel Cook, ein wahres Game-Design-Genie und ein exzellenter Artist. Vor SteamBirds hätte ich mir vermutlich nie träumen lassen, mal mit einem eigenen »Game Designer« zusammenzuarbeiten, weil ich die grundlegende Denkweise immer einfach schon vorausgesetzt hatte. Ich habe Dan vor allem als Artist angeheuert – jetzt denke ich, dass ich niemals wieder ein Spiel ohne seinen qualifizierten Rat entwickeln könnte.
- xx% gehen an Danny Baranowsky für sein musikalisches Genie (davon wiederum bezahlt er Jordan Fehr für die Sound-Effekte). Ich sehe eine Menge erschrockener Blicke, wenn ich das hier erzähle, und ich kann auch verstehen warum. Ich bin es ebenfalls gewohnt, nur einen Bruchteil für Audio auszugeben. Aber Dannys Dienste (Stichwort »Perpetual Unlimited Music«!) und das Endresultat waren einfach absolut atemberaubend. Ich glaube an das Gameplay als Alleinstellungsmerkmal meines Spiels, aber die Musik bringt das ganze Produkt noch einmal auf einen neuen Level – es hat sich gelohnt!

Die Auktion

Die Interessenten konnten ab dem 30. Januar 2010 ihre Gebote für SteamBirds bei Flash Game License abgeben. Ich habe mich durch die Sales-Statistiken von Flash Game Licenses informiert, um einen besseren Überblick zu gewinnen. Offenbar schien der Januar aber der schlechteste Monat zu sein, direkt gefolgt vom Februar. Am liebsten hätte ich also bis in den späten Sommer gewartet, ich wollte aber auf gar keinen Fall den Release des Spiels weiter verzögern.

Am Ende haben sieben Interessenten über Flash Game License für das Spiel geboten, ein weiteres Dutzend hat mir Angebote quasi inoffiziell per Mail unterbreitet. Wir sind also einige Optionen parallel zum Bieterprozess durchgegangen. Diese Angebote waren in der Regel so komplex, dass es nicht angebracht schien, sie an Flash Game License weiterzugeben. Die sieben aktiven Bieter gaben 27 Angebote während eines zehntätigen Kampfes um SteamBirds ab. Startgebot waren 500 Dollar, der Preis kletterte aber rasend schnell auf die 6.000-Dollar-Marke.

Andy Moore arbeitet eng mit **Flash Game License** zusammen, die Flash-Entwickler in Kontakt mit einer großen Bandbreite an Geldgebern bringen.

Hier ein kleiner Abriss des Bietvorgangs, inklusive der Mail-Angebote:

- 30. Januar: 500 Dollar:
  Das eröffnende »LOL«-Angebot
- 31. Januar: 1.500 Dollar
- 1. Februar: 2.000 Dollar
- 1. Februar: 6.000 Dollar: Ein großer Sprung gleich zu Beginn. Es gab keine Angebote zwischen 2.000 und 6.000 Dollar.
- 4. Februar: 8.000 Dollar:
  Ein Angebotskrieg entbrennt.
- 4. Februar: 10.000 Dollar
- 5. Februar: 16.000 Dollar
- 6. Februar: 20.000 Dollar
- 7. Februar: 21.000 Dollar
- 8. Februar: 22.000 Dollar
- 9. Februar: **25.000 Dollar:**
  Dieses Angebot habe ich angenommen.
- 12. Februar: 30.000 Dollar
- 12. Februar: 35.000 Dollar
- 15. Februar: 45.000 Dollar

Ich habe die Auktion um den 16. Februar herum geschlossen und Dan bei Armor Games informiert, dass sein Angebot am Ende den Zuschlag gekriegt hat. Unser Gespräch verlief in etwa so:

Dan: »Aber ... aber warum? Warum würdest du ein so lächerlich niedriges Gebot annehmen?!«

Ich: »Das kann ich dir sagen ...« Daraufhin erklärte ich ihm die ganze Situation in beinahe erschreckend ruhigem Ton.

Die hochpreisigen Deals hatten einige wirklich fiese Knebelklauseln in ihren Licensing Terms. Hier mal eine kleine Aufstellung, um nur ein paar der gängigen Forderungen zu nennen:

Vorkaufsrecht für ein Sequel

Ich weiß zwar, dass ein Vorkaufsrecht nicht wirklich unter dem Begriff »Knebel« läuft, aber es verschafft mir trotzdem ein ungutes Gefühl in der Magengegend. Was, wenn ich schlechte Erfahrungen mit dem Publisher gemacht habe? Ich will mich nicht durch irgendetwas gefesselt fühlen. Wenn sich der Publisher gut verhalten hat, bin ich bei einem Sequel gern bereit, wieder

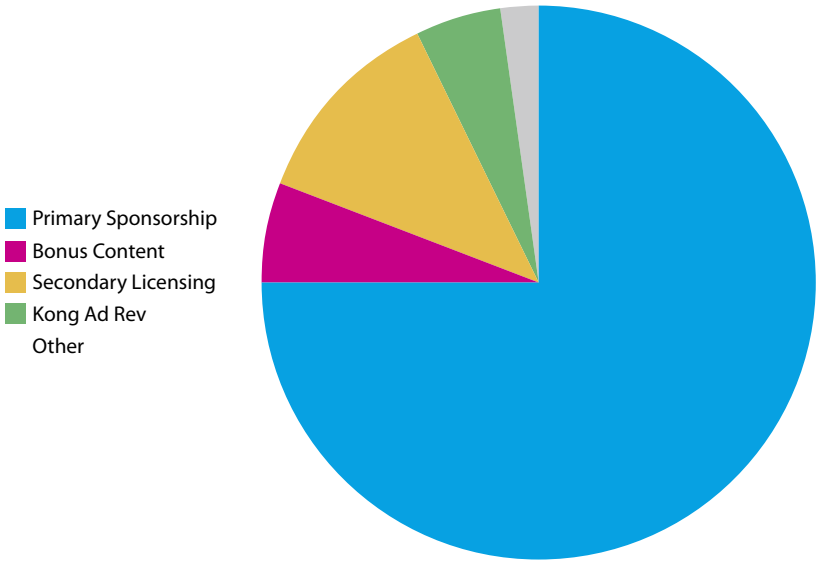


Abbildung 1: Die **Einnahmen** einen Monat nach dem Release von SteamBirds.

mit ihm zu arbeiten. Warum sollte man aber so eine Freundschaft schriftlich fixieren müssen?

Keine Sequels oder Spin-offs; sowie keine Verwendung von existierenden oder ähnlichen Assets für ein Jahr

Ich habe so einige große Pläne für die Zukunft, und die SteamBirds-IP ist sprichwörtlich reif, gepflückt zu werden. Besonders jetzt, wo sich der Erfolg einstellt, kann ich dies einfach nicht ignorieren. Und ja, ich verstehe natürlich, dass man den Raum zwischen zwei Titeln wahren sollte, ich möchte den Markt nicht mit meinen Produkten übersättigen. Aber ich will auch nicht zwangsweise für ein komplettes Jahr an meiner Arbeit gehindert werden.

Dieser Deal schließt außerdem eine finanzielle Vorausleistung für ein Spin-off ein, das eine Reihe Features XYZ enthält.

Diese Angebote waren natürlich verlockend. Am Ende hätte das Extrageld aber nicht die Arbeit an neuen Features aufgewogen, sodass ich von derartigen Deals absah.

Dauerhafte und exklusive Lizenz, keine sekundären Verkäufe oder Anzeigenerlöse

Diese Klausel geht durchaus in Ordnung – aber nur, solange man den Einnahmenverlust durch den Deal ausgleichen kann.

Wie ich die Wahl für einen Sponsoren traf

Um mir Orientierung bei all den Angeboten zu verschaffen, habe ich mir einen »Value Scale« ausgedacht. Ich habe versucht zu antizipieren, wie viel mein Spiel wohl nach Zweitverkäufen, Werbeeinnahmen und Ähnlichem erlösen könnte. Dann habe ich noch etwaige Leistungsboni, Preisgelder und andere Spezialfälle eingerechnet. Diese Werte habe ich schließlich auf den Basispreis des Angebots addiert. Nach der Rechnung war das Angebot von Armor Games rund 30.000 bis 40.000 Dollar wert, unabhängig vom direkten Preis von 25.000 Dollar.

Unglücklicherweise führen die strikten Lizenzvereinbarungen bei den großen Publishern zu keinem derartigen Wertaufschlag. In Wahrheit kam es bei einigen sogar zu negativen Werteverschiebungen, wenn längere Laufzeiten und Exklusivdeals im Spiel waren.

Am Ende blieben mir zwei ähnliche Deals, die ich ernsthaft in Erwägung zog. Sie hatten vergleichbare Vorgaben und Armor Games war sogar etwas hinter seinem Konkurrenten, was den »Value Scale« anbelangte. Trotzdem habe ich mich für sie entschieden, aus einem ganz ausschlaggebenden Grund: Sympathie.

Es war einfach fantastisch, mit Dan McNeely zu arbeiten. Ganz zu Beginn des Bieterprozesses haben wir uns zusammengesetzt und einige Dinge besprochen – dabei sogar einiges, das überhaupt nichts mit Spielen zu tun hatte. Diverse Male hat er mir empfohlen, doch lieber andere Deals zu akzeptieren, die besser für mich seien. Er hat ständig versucht, neue Vereinbarungen auszuarbeiten, hat andere Publisher vorgeschlagen, die ich eventuell mal ansprechen sollte, und zu denen er einen guten Draht hatte. Und das Beste ist: Er hat seine Lizenzbedingungen stets sehr offen gehalten, mit erstaunlich wenigen Restriktionen.

Die meisten Publisher sagen: »Dieser Deal ist der beste für dich, lieber Entwickler!« Das klingt immer irgendwie nach Gebrauchtwagenverkäufer. Dan hingegen hat mein Vertrauen gewonnen. Und ich habe wirklich das Gefühl, dass er sich um mein Wohl sorgt. Er muss dies nicht erst aussprechen, ich glaube es ihm auch so.

Publisher sind deutlich mehr als nur Geldmaschinen. Sie sind Partner, die deine Zukunft als Entwickler mitbestimmen. Dans Angebot zu akzeptieren hat vermutlich eine lange und fruchtbare Beziehung zur Folge. Ich vertraue ihm, die besten Deals für mich zu finden – hoffentlich wieder mit ihm selbst!

Es lässt sich nur schwer in Zahlen ausdrücken, aber diesen Mehrwert lasse ich mir gern etwas kosten. Es war kaum zu glauben, wie viele Sponsoren mit einer extrem aggressiven Haltung und Beleidigungen auf eine Absage reagiert haben. Einige von ihnen wollen gar nicht mehr mit mir sprechen und antworten auch nicht mehr auf Mails. Sie fühlen sich übermächtig gegenüber Entwicklern und erschaffen eine »Du wirst dich uns schon beugen!«-Atmosphäre. Jetzt mal im Ernst: Das ist doch wirklich alles andere als professionell!

Die Vertragsbedingungen

Dan hat mir freundlicherweise erlaubt, an dieser Stelle die Details des primären Publishing-Vertrags zu nennen (natürlich mit den bereits erwähnten, kleinen Anpassungen):

- Vorauszahlung von 25.000 Dollar für SteamBirds
- Bonus von 2.000 Dollar für Bonus-Missionen, exklusiv für Armor Games
- Implementierung der Armor Games AGI (High Score Interface)
- Exklusiv-Periode von einer Woche bei Armor Games (ad-free)
- Darauffolgend eine Woche »Viral Version«-Distribution, Werbung erlaubt
- Wieder darauffolgend sind Zweitverkäufe erlaubt, inklusive Werbung

Dan hat mir keinen ellenlangen Vertrag geschickt, den ich unterschreiben sollte. Er war völlig damit zufrieden, ein paar Mails mit den Details auszutauschen. Er war wirklich flexibel und hat jeden Vorschlag einzeln abgewogen. Im Gegenzug für seine Großzügigkeit habe ich ihm eine längere Exklusivperiode gegeben, als er eigentlich gewünscht hatte. Diese Beziehungen beruhen immer auf Gegenseitigkeit, und nichts ist dabei in Stein gemeißelt.

Die harten Zahlen

Es ist Zeit für eine Abrechnung (siehe **Abbildung 1**). Im Tortendiagramm, das die Einnahmen einen Monat nach Release zusammenfasst, läuft unter »Other« die geballte Kraft von CPMStar- und Mochi-Advertising. Hinzu kommt etwas Geld aus Awards und Performance-Meilensteinen. Alles in allem hat SteamBirds etwa 34.000 Dollar eingebracht, was ziemlich genau meine ursprüngliche Erwartung trifft. Wenn man jetzt Flash Game License und das SteamBirds-Team abzieht, bleiben 15.000 Dollar für meine eigene Geldbörse nach einem Monat Vollzeit-Arbeit übrig. Selbst wenn man das über die gesamte Entwicklungszeit – etwa fünf Monate – verteilt, bleiben immer noch 3.000 Dollar pro Monat übrig – ein ganz ordentliches Gehalt.

Natürlich kommen später auch noch steigende Werbeeinnahmen hinzu, die vielleicht einmal ganz gut etwas einbringen. Und es gibt sicherlich auch noch einige weitere Licensing Deals, die man abschließen könnte.

Game Performance Tracking

Großen Dank an Ben von SWFStats! Seine äußerst einfach zu implementierende API ist ziemlich genial und wirklich beeindruckend. Die Daten laufen exakt parallel zu denen anderer Collection Services, das spricht für die Stabilität und die Authentizität. **Abbildung 2** zeigt einen Überblick über den Traffic im ersten Monat.

Dazu einige Fakten zu den ersten Wochen nach Release:

- 3.500.000 Game Loads
- 24.000.000 Levels gespielt
- 26:10 Minuten durchschnittliche Spielzeit
- 1.000 Seiten hosten das Spiel

Es gibt eine statistisch zu vernachlässigende Gruppe Spieler, die SteamBirds weniger als zwei Minuten lang spielt. Die offiziellen Statistiken sprechen im Fall der Retention Rate von 100%. Das heißt, dass jemand, der den Titelbildschirm sieht, im Durchschnitt auch zwei Levels spielt.

- 53% der Spieler spielen mehr als fünf Levels.
- Nur 1,3% der Spieler benutzen den Mute-Button (ohne das Spiel wieder zu »unmuten«).
- Nur 3% der Spieler sehen sich die Credits an.

Und zu guter Letzt noch ein paar »Per Site«-Traffic-Stats:

- ArmorGames.com zieht den meisten Traffic.
- Kongregate.com ist weit abgeschlagen auf dem zweiten Platz, mit weniger als einem Drittel Views.
- Eine chinesische Seite hat das Spiel gleich an Tag 4 geklaut und ist mittlerweile auf Platz 3, allerdings mit geringem Nennwert.
- Der Rest meiner 1.000 Seiten hat jeweils weniger als 50.000 Views.

Ich hoffe, diese Zahlen und Erfahrungswerte konnten Ihnen ein wenig weiterhelfen.

Andy Moore

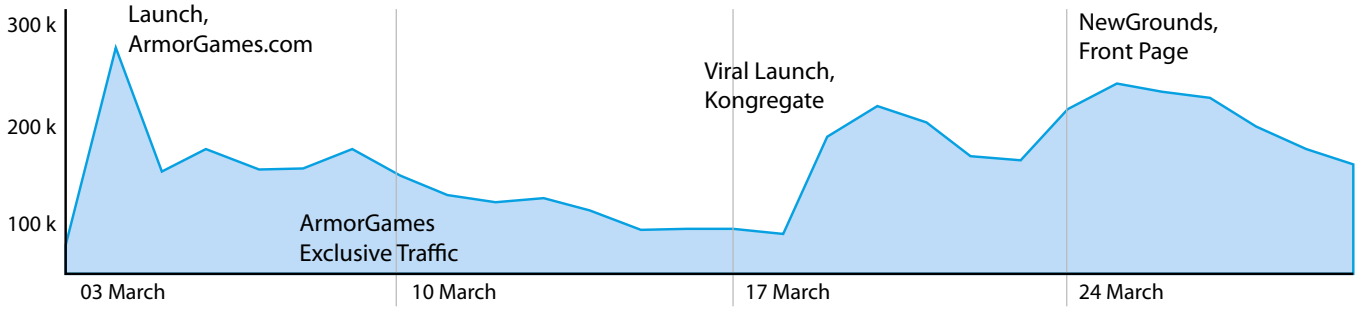


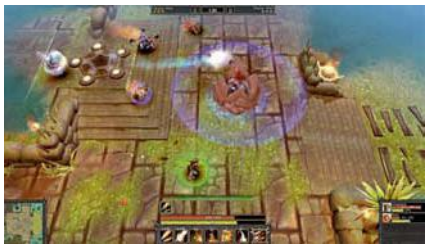
Abbildung 2: So verteilt sich der **Traffic** auf die größten Anbieter-Seiten.

Jung, weiblich, CEO
Im Gespräch mit Tau Petersson

Tau Petersson hat als Studentin das vielversprechende Online-Spiel Bloodline Champions beim Publisher Funcom untergebracht. Wir sprechen mit ihr über ihren bemerkenswerten Weg in die Spieleindustrie.

Tau Petersson

ist CEO der Stunlock Studios.



Das Online-Spiel **Bloodline Champions** erinnert auf den ersten Blick an das DotA-Spielprinzip. Statt individualisierbarer Charaktere und Stufenaufstiege setzt das PvP-Spiel aber absichtlich auf identische Helden und eine klassische Shooter-Steuerung.

Die Stunlock Studios sind aus einem Projekt der **University of Skövde** hervorgegangen. In drei Ausbildungszweigen werden die Studierenden entweder zu Programmierern, Graphic Artists oder Designern ausgebildet.

Making Games Kannst du uns erzählen, wie die Stunlock Studios entstanden sind und wie du zum CEO wurdest?
Tau Petersson Es hat alles als Studentenprojekt an der University of Skövde begonnen. Die Universität hat drei Ausbildungszweige im Game Development: einen für Programmierer, einen für Graphic Artists und einen für Designer. Im dritten und letzten Jahr des Studiums hat sich unser Team zusammengefunden. Im Rahmen eines Game-Development-Projekts entstand die Idee für Bloodline Champions, und die Leute schienen das Konzept wirklich zu mögen. Also entschieden wir uns, das Spiel weiterzuentwickeln. Ich stieß allerdings erst ein klein wenig später hinzu, um den Project Lead und das Interface Design zu übernehmen. Allerdings hat das Project Leading am Ende etwas Überhand genommen, also wird das Interface jetzt von einem Kollegen designt. Als das Projekt schließlich Form annahm, wurden wir unter anderem bei den Swedish Game Awards mit zwei Preisen belohnt, sogar für das »Beste Spiel 2009«. Von da an war für uns klar, dass wir das Projekt professionell als Unternehmen fortführen wollen. Wir haben schließlich mit »The Game Incubator« gesprochen, einer Startup-Hilfe, die mit unserer Universität zusammenarbeitet. Sie haben uns zum Beispiel in Sachen Büros und Equipment weitergeholfen. Und so wurde ich schließlich CEO und habe mich um das Organisatorische und die Kontakte zum Incubator gekümmert.

Making Games Wie habt ihr es dann so schnell geschafft, für euer erstes Spiel einen Publishing-Deal mit Funcom zu bekommen?
Tau Petersson In der Zeit als Studententeam haben wir uns natürlich nicht um Publisher bemüht. Das hat sich dann aber nach den Swedish Game Awards geändert. Der Incubator hat uns geholfen, einen Pitch aufzubauen und er hatte auch schon einige direkte Kontakte zu Publishern. Im März 2010 haben wir dann das »Level Up«-Programm und damit freien Eintritt für die Game Connection in San Francisco gewonnen. Das gab uns die Möglichkeit, unser Spiel bei den großen Publishern vorzustellen. Die Game Connection ist wie Speed Dating. Wenn du Glück hast und viele Publisher mit dir sprechen möchten, kannst du im Halbstundentakt einer Menge Leute dein Projekt zeigen. Wir haben mit ungefähr 25 Firmen gesprochen, das war eine ganz schön hektische und aufregende Zeit. Und tatsächlich war unser allerletztes Meeting mit Funcom. Fast hätte es gar nicht mehr mit dem Termin hingehauen, denn der Funcom-Mitarbeiter vor Ort war bereits vollkommen ausgebucht. Also habe ich noch schnell seinen Kollegen per Mail angeschrieben und wir haben wirklich noch einen Termin gekriegt. Na ja, und es ist ziemlich gut gelaufen (lacht).

Making Games Was macht Bloodline Champions so besonders, vor allem im direkten Vergleich zu ähnlichen Titeln wie etwa DotA?
Tau Petersson Bloodline Champions konzentriert sich weit mehr auf reines PvP, als dies etwa bei DotA der Fall ist. Die WASD-Steuerung und das aktive Aiming mit der Maus machen das Spiel noch etwas actionlastiger. Hinzu kommt, dass bereits alle Skills gleich zu Beginn zur Verfügung stehen, die Spieler müssen also keine Zeit investieren und ihre Helden hochleveln oder mit Gegenständen ausrüsten. Uns ist wichtig, dass man direkt in ein Spiel springen kann und dass jeder dieselben Voraussetzungen für einen Kampf mitbringt.

Making Games Für welches Geschäftsmodell habt ihr euch bei eurem Spiel entschieden?
Tau Petersson Wir nutzen ein Free-to-play-Modell, bei dem die Spieler kostenlosen Zugang zu einigen unserer Heldenklassen, den Bloodlines, erhalten. Welche das sind, entscheidet eine wöchentliche Rotation. Hinzu kommt ein Item-Shop, in dem über Micro Transactions Gegenstände erworben werden können, die allerdings niemals die Spielbalance in Gefahr bringen. Wir bieten zudem einige Bundles an, mit denen die Spieler etwa neue Outfits, Waffenskins oder auch weitere Slots in der Teamaufstellung freischalten können. Die Spieler sammeln zudem Blood Coins, mit denen sie nach und nach auch noch die weiteren Bloodlines freischalten. Diese

Heldenklassen sind der Mittelpunkt des Spiels und beeinflussen das Gameplay am stärksten. Uns ist daher sehr wichtig, dass keinem Spieler die Bloodlines verwehrt bleiben.

Making Games Ihr hattet bestimmt noch nicht all diese Details ausgearbeitet, als ihr das erste Mal auf Funcom traft. Was habt ihr denn auf der Game Connection vorgelegt, um euren zukünftigen Publisher zu überzeugen?
Tau Petersson Es ist einfach wichtig, ein Spiel zu haben, das sich gut anfühlt und natürlich auch gut aussieht. Bloodline Champions hat in gewisser Weise schon für sich selbst gesprochen. Wir haben aber hinzugefügt, welche Features wir unbedingt noch einbauen möchten. Und wir hatten natürlich auch den Incubator im Rücken, was das Risiko minimierte und wodurch wir im Vergleich zu anderen Studios vielleicht etwas sicherer erschienen.

Making Games Es reicht also nicht, allein mit einem Konzept und einer Powerpoint-Präsentation vorstellig zu werden?
Tau Petersson Auf keinen Fall! Wir hatten bereits eine vollspielbare Version von Bloodline Champions mitgebracht. Nur so kann man sich von den unzähligen anderen Projekten abheben, aus denen die Publisher wählen müssen. Bei einem Prototypen können sie selbst Hand anlegen und sagen »Hey, das ist cool und könnte wirklich funktionieren«.

Making Games Du hast schon mehrfach den »Incubator« erwähnt und wie er euch geholfen hat. Die schwedische Games-Branche scheint ja sehr Startup-freundlich zu sein.
Tau Petersson Der Incubator hat uns schon sehr weitergeholfen. Er bemüht sich auch, für uns weitere öffentliche Mittel zu beschaffen. Aber im Allgemeinen ist die schwedische Industrie wohl nicht unbedingt besonders Startup-freundlich. Während des Studiums haben wir vielmehr ganz normale Darlehen erhalten, die uns bei der Finanzierung geholfen haben. Und dann sind wir auf Funcom getroffen, die uns nun bei der Entwicklung unter die Arme greifen.

Making Games Was waren denn für dich persönlich die größten Herausforderungen bei der Unternehmensgründung?
Tau Petersson Die größte Herausforderung – natürlich auch für mich persönlich – war tatsächlich die Publisher-Suche. Wir haben zwar darüber nachgedacht, wie wir das Ganze als Indie aufziehen würden, aber ein Spiel wie Bloodline Champions wäre ohne großen Publisher im Rücken nicht wirklich machbar gewesen. Es ist ein echt hartes Business und wirklich schwierig, einen Publisher auf sich aufmerksam zu machen. Es gibt unfassbar viele Spielkonzepte dort draußen und die Publisher müssen sich eine Menge anschauen und dann bewerten, was Zukunft haben könnte und was nicht. Sie kriegen sicherlich einen Haufen Mails, die sie aussieben müssen. Da kann es auch schnell mal passieren, dass ihnen etwas Spannendes entgeht.

Making Games Es gibt nicht viele weibliche CEOs in der Branche. Liegt das am Chauvinismus der Männer oder bewertest du die Chancen auf Führungspositionen für Frauen gleich?
Tau Petersson Meiner Erfahrung nach ist es so, dass Frauen in Spiele-Unternehmen meist auch in Management-Positionen arbeiten. Ich denke also, dass der Weg zum CEO prinzipiell gut möglich ist, denn genau da wollen sie ja hin. Es gibt heutzutage nichts, was eine Frau von einer hohen Position ausschließen würde.

In unserem Team aus 14 Leuten bin ich im Moment die einzige Frau, aber ein gemischtes Team trägt auch zur guten Atmosphäre bei. Ich arbeite zwar als CEO, aber ich kann mir Frauen auch durchaus in anderen Positionen vorstellen.

Making Games Du sagtest, dass ihr mit vielen Publishern auf der Game Connection gesprochen habt und diese euer Projekt interessant fanden. Wie habt ihr euch dazu entschieden, mit Funcom weiterzumachen?
Tau Petersson Wir haben schon vorher mit einigen kleineren Publishern gesprochen. Aber wir hatten das Gefühl, dass unser Projekt nicht wirklich gut bei ihnen aufgehoben gewesen wäre. Auf der Game Connection haben wir dann auch eine Reihe größerer Publisher getroffen, die meisten von ihnen waren jedoch nicht auf Online-Spiele spezialisiert. Wir hatten daher das Gefühl, dass die Prozesse dort etwas träge sein würden. Und wir waren schließlich auch noch unerfahren in all diesen Abläufen und brauchten einen Partner, der sich bereits sehr gut im Online-Bereich auskennt. Mit Funcom haben wir so einen Publisher gefunden. Sie kennen den Online-Markt, das Marketing, und durch ihre räumliche Nähe zu unserem Studio funktioniert auch die Kommunikation sehr gut. Es wäre ein bisschen eigenartig für uns gewesen, zum Beispiel einen Publisher aus den USA zu haben. So stehen wir wirklich immer im direkten Kontakt, egal welche Frage wir gerade haben. Alle zwei Wochen kommt auch jemand von Funcom zu uns ins Büro, und ab und an sind wir in Oslo, um uns abzusprechen und das Projekt weiter voranzutreiben.

Making Games Wie sieht eure Roadmap für die nächsten Monate aus?
Tau Petersson Im Moment konzentriert sich unser gesamtes Team darauf, neue Features zu entwickeln und zu testen. Nach dem Release werden wir immer weitere Bloodlines hinzufügen, was eine Menge Zeit in Anspruch nimmt. Und danach werden wir vermutlich einige unserer Leute abziehen, die sich Konzepte für ein neues Spiel ansehen werden.

Making Games Wir wünschen euch viel Erfolg! Vielen Dank für das Interview, Tau!

Interview: Yassin Chakhchoukh

Der **Game Incubator** unterstützt schwedische Start-ups bei der Suche nach Geldgebern, hilft bei Fragen rund um die Finanzierung und stellt ein voll ausgestattetes Studio für die Entwicklung zur Verfügung.

Die Stunlock Studios haben über das »Level Up«-Programm freien Zugang zur **Game Connection** in San Francisco erhalten. Dadurch konnten sie mehr als 25 Publisher treffen und am Ende auch Funcom für Bloodline Champions begeistern.

Die **Bloodlines** bilden die fest vorgegebenen Heldenklassen des Spiels. Jede hat sieben Talente, die nicht von den Spielern angepasst werden können.

Wie Entwickler den rechten Weg leuchten
Beeinflussung von Spielern

Richard Winterstetter beschäftigt sich im Studium mit Mechanismen der Spielerbeeinflussung, Möglichkeiten der Immersion und den Auswirkungen auf das Game und Level Design. In seinem Essay beschreibt er, warum dieses Thema für die Spieleentwicklung von so großer Wichtigkeit ist.

Richard Winterstetter

ist Game-Design-Student an der Mediadesign Hochschule in München.

Richard arbeitete bereits während seines Studiums an dem 2D-Jump&Run Grounded mit, das mit dem zweiten Platz beim Gamesload Newcomer Award 2009 ausgezeichnet wurde. Während der Entwicklung kümmerte er sich unter anderem um das Gegnerdesign. Derzeit beschäftigt er sich ausführlich mit dem Leveldesign sowie der Narration in Spielen und wird sein Studium voraussichtlich im September 2011 abschließen.

WWW
makinggames.de
- Stimmung im Rollenspiel Two Worlds 2 erzeugen

Game Designer zu sein ist ein Fluch! Je mehr man über Spielerführung lernt, desto weniger funktioniert sie bei einem selbst. Diese Erfahrung musste ich im Rahmen meines Game-Design-Studiums an der Mediadesign Hochschule in München machen. Dort beschäftige ich mich ausgiebig mit den gängigen Methoden der Spielerführung. Ich stelle vor allem fest, dass »spielerführende« Mechanismen immer seltener funktionieren, wenn ich selbst spiele. Meist enttarne ich sie sofort und die Wirkung verfliegt. Darunter leidet der Spielspaß und ich begann mich nach einiger Zeit zu fragen, ob man den Spieler nicht unbewusst auf psychologischer Ebene beeinflussen kann, um eben diesen Effekt zu vermeiden.

Die Spieler selbst haben sich natürlich stetig weiterentwickelt – Methoden zur Spielerführung, die vor zehn Jahren noch funktionierten, werden heute oft erkannt und nicht selten bewusst ignoriert. Deshalb müssen sich Entwickler immer subtilere Methoden einfallen lassen, den Spieler durch einen Level zu führen. Wenn der Spieler nie das Gefühl hat, gelenkt oder zu bestimmten Entscheidungen gezwungen zu werden, dann ist dies ein großer Schritt auf dem Weg, eine immersive Erfahrung zu schaffen. Aber auch über das simple Lenken des Spielers hinaus können Entwickler durch subtile Beeinflussung das Spielerlebnis verbessern, beispielsweise durch das geschickte Einbauen von Storyelementen, die Spieler und Spielfigur zusammenführen. Man muss also gezielt mit psychologischen Tricks sowie unbewussten Reizen und Motiven arbeiten, um den oben genannten Effekt auf den Spieler zu erzielen. Auf der Suche nach dem verloren gegangenen Spielspaß habe ich mich ausführlich mit diesem Thema auseinandergesetzt und dabei einige interessante Game-Design-Merkmale herausgefunden.

Den Spieler lenken

Zunächst möchte ich einen grundlegenden Mechanismus definieren, auf den alle in diesem Artikel behandelten Beispiele aufbauen: Der Spieler sollte stets das Gefühl haben, jede Entscheidung aus freiem Willen zu treffen. Hat er auch nur für einen Augenblick das Gefühl, dass die Entwickler ihn in eine bestimmte Richtung drängen wollen, wird er diesem Vorhaben mit hoher Wahrscheinlichkeit mit Ablehnung begegnen. Niemand mag es, manipuliert zu werden, aber genau das tun Entwickler in der Regel mit ihren Spielern. Es wird die Illusion geschaffen, sich frei in einer virtuellen Welt zu bewegen, obwohl man im Endeffekt »nur« die Orte besuchen und die Handlungen ausführen darf, die der Entwickler dem Spieler zugesteht. Selbst in MMOs oder Spielen mit riesigen Welten (etwa Fallout 3) wird der Spieler nie eine Handlung ausüben dürfen, deren Rahmenbedingung nicht zuvor festgelegt wurde.

Damit ihm das Spielen trotzdem Spaß macht, gibt ihm der Entwickler das Gefühl der Entscheidungsfreiheit. Hierzu ein Beispiel: Der Spieler soll an einer Wegkreuzung den rechten Pfad

Atmosphärische Details können auch für die Spielerführung zweckentfremdet werden. In **Dead Space** wird das Licht etwa auch genutzt, um Besonderheiten in den Levels hervorzuheben.

einschlagen. Der einfache Befehl »Geh nach rechts« wäre an dieser Stelle Ausdruck schlechten Designs. Stattdessen sollten Entwickler mit Points of Interest (POIs) den Eindruck beim Spieler erwecken, dass der rechte Weg der richtige, der interessante, der bessere Weg ist. Natürlich könnte man auch an Stelle der Kreuzung einfach den Weg nach rechts abbiegen lassen, aber das würde wiederum die Illusion der freien Welt zerstören. Im Folgenden werde ich auf die unterschiedlichen Varianten von POIs eingehen, die Entwickler je nach Situation einsetzen können.

Es werde Licht

Das Lighting eines Levels ist nicht nur ein notwendiger und wichtiger Teil des Level Designs an sich, sondern kann gezielt zur Spielerführung eingesetzt werden. Aus einem unbewussten Drang heraus folgt man in einem finsteren Level eher einer Lichtquelle, als im Dunklen herumzuirren. Wenn man diese Tatsache auf das vorherige Beispiel anwendet, könnte der Level Designer den rechten Weg einfach besser ausleuchten oder in einiger Entfernung ein leuchtendes, helles Objekt als eine Art »Leuchtturm« platzieren – wie etwa in den zahlreichen düsteren Wald-Passagen im Actionspiel Alan Wake. Die meisten Spieler werden dann, wenn sie an einer Wegkreuzung angelangt sind, instinktiv den helleren Weg wählen, zum einen aus dem bereits beschriebenen Reflex heraus, zum anderen, weil der rechte Weg wesentlich interessanter wirkt. Der Spieler glaubt also, aus freiem Willen heraus nach rechts zu gehen, obwohl der Entwickler genau das von ihm will.

Die Macht der Farben

Ebenso wie mit dem Licht verhält es sich auch mit unterschiedlichen Farben. Wenn man einen POI mit Farben erzeugen möchte, sollte man unbedingt darauf achten, dass diese mit ihrer Umgebung in Kontrast stehen. In einer bunten Comicwelt fällt dem Spieler ein knalliges Rot eben nicht so stark ins Auge wie beispielsweise in der sterilen Welt eines Mirror's Edge.

Es kann ebenfalls mit der richtigen Positionierung von farbigen Objekten im Level dafür gesorgt werden, dass der Spieler automatisch den richtigen Weg einschlägt. Dazu muss ihm die spielerführende Funktion einer Farbe erst vermittelt werden. Am einfachsten lässt sich das erreichen, indem man dem Spieler Zeit gibt, die Farbe wahrzunehmen und sie gewissermaßen unbewusst als Signalfarbe abzuspeichern.

Wendet man dies auf unser Beispiel an, würde der Leveldesigner etwa auf dem Weg zur Kreuzung und anschließend auf dem rechten Pfad Objekte einer bestimmten Farbe verteilen. Während sich der Spieler auf die Kreuzung zubewegt, nimmt er die Farben mehr oder minder bewusst wahr. An der Kreuzung angelangt, wird er im Idealfall die farbigen Objekte auf dem rechten Weg deutlich erkennen und sich automatisch den bereits vertrauten Assets mit derselben Farbgebung zuwenden. Auf diese Art

In Half-Life 2 werden **POIs** eingesetzt, um bestimmte Wege optisch interessanter zu gestalten.

Besonders in **Alan Wake** wird das Spiel mit dem Licht gekonnt für die Spielerführung eingesetzt. Leuchtende Objekte zeigen bereits aus der Ferne, wo der nächste sichere Hafen liegt.

kann der Spieler mit dem gezielten Einsatz von farbigen Objekten durch komplexe Räume geführt werden, ohne dass der Spielfluss verloren geht oder dass sich der Spieler zu sehr gelenkt fühlt. Neben Licht und Farbe lassen sich auch mit geschickt platzierten Sounds oder Animationen POIs generieren. In der Praxis werden oft mehrere Faktoren kombiniert, um sicherzustellen, dass die POIs auch tatsächlich funktionieren.

Immersion und Emotion

Die Beeinflussung des Spielers kann weit mehr erreichen, als ihn »nur« durch den Level zu lenken. Während POIs Tools sind, mit denen der Level-Designer direkt arbeiten kann, ist es auch durch geschickte Inszenierung und die Narration möglich, den Spieler zu beeinflussen. Jedoch sind diese weiterführenden Tools ein zweischneidiges Schwert. Richtig angewandt und rezipiert entfalten sie eine weit größere, mächtigere und differenziertere Wirkung auf den Spieler als der bloße Einsatz von POIs. Jedoch hängt ihre Wirkung auch davon ab, inwieweit

In der sterilen Welt eines **Mirror's Edge** fällt die rote Signalfarbe zur Spielerführung besonders gut auf.

sich der Spieler auf das Spiel einlässt und sich von der erzählten Geschichte mitreißen lässt.

Es ist eine besonders im Horror-Genre gängige Vorgehensweise, den Spieler durch geschicktes Ansprechen von Reflexen und Urängsten in eine unruhige Grundstimmung zu versetzen, was ihn wiederum anfälliger für geskriptete Schockmomente macht.

Was die Beeinflussung der emotionalen Grundstimmung des Spielers angeht, so kann man mit vielen Elementen wie Licht und Schatten, Lichtfarbe, Sounds und Musik arbeiten. Wie schon bei der Spielerführung erzielt aber erst die richtige Kombination der einzelnen Faktoren das optimale Ergebnis. Neben der emotionalen Grundstimmung lässt sich auch der Stresslevel und die Art, wie der Spieler den jeweiligen Level wahrnimmt, beeinflussen.

Wenn man dem Spieler etwa vermittelt, dass ein gewisser Zeitdruck herrscht, wird er anders durch den Level wandern als unter normalen Umständen. Sein Stresslevel steigt, und er wird meist dem durch Licht und Farben klar erkennbar ausgearbeiteten Hauptweg folgen. Dabei entwickelt er eine Art Tunnelblick, der verhindert, dass sich der Spieler mit unwichtigen Leveldetails beschäftigt. Dieses Ergebnis lässt sich auf mehrere Arten erreichen, beispielsweise mit einem eingeblendeten Countdown. Die (meist negativen) Konsequenzen, die bei Versagen eintreten, werden dem Spieler durch den Zähler eindeutig und dauerhaft vor Augen geführt, folglich wirkt sich dies emotional auf sein Spielverhalten und seinen Stresslevel aus. Ästhetischer ist es allerdings, durch die Inszenierung und den geschickten Einsatz von Storyelementen dieselbe emotionale Grundlage beim Spieler zu schaffen.

Gezielt Emotionen nutzen

Wenn der Entwickler beispielsweise eine Verfolgungsjagd inszeniert, bei der der Spieler einen flüchtenden NPC verfolgen muss, und ihn immer wieder einen Blick auf den Gejagten erhaschen lässt, so erreicht er beim Spielenden mit hoher Wahrscheinlichkeit dieselbe Wirkung und dasselbe Verhalten wie mit Hilfe des eingeblendeten Countdowns. Wichtig dabei ist, dass der Spieler stets das Gefühl hat, den Verfolgten einholen zu können, wenn er nur schnell genug agiert. Ein gutes Beispiel dieser Technik liefert etwa das Actionspiel »Kane & Lynch 2:

In **Kane & Lynch 2: Dog Days** verfolgt der Spieler einen NPC. Dieser gerät nie ganz aus dem Blickfeld, sodass der Spieler stets angetrieben wird, ihn weiterzujagen.

Dog Days«: Im ersten Kapitel verfolgt der Spieler einen Flüchtigen. Immer wenn er einen neuen Abschnitt betritt, wird ihm durch ein kurzes Auftauchen des Verfolgten vermittelt, wohin er zu laufen hat. Instinktiv rennt der Spieler los, um den Flüchtigen einzuholen. Er wird durch den Glauben angetrieben, den NPC einholen zu können und im Falle des Scheiterns eventuell den »Game over«-Screen sehen zu müssen. Dass es sich bei den Einblendungen des Flüchtenden um geskriptete Elemente handelt und weder das Einholen noch das »Entkommen lassen« möglich ist, wird dem Spieler in diesem Moment nicht klar.

Eine derartige Fluchtszene erhöht also die Immersion, während ein eingeblendeter Countdown schnell zur Frustfalle werden kann. Er besitzt keinerlei spielerführende Wirkung, die tickende Uhr setzt den Spieler ausschließlich als ein abstraktes Interface-Element unter Druck.

Die Motivation des Spielers deckt sich im obigen Beispiel kurzzeitig mit der des Spielcharakters: Es gilt, den Flüchtenden zu fangen. Die Königsklasse der Spielerbeeinflussung ist es also, wenn sich Motivation des Spielers und Motivation des Spielcharakters durch eine starke Identifikation dauerhaft decken. Die Quest des Charakters wird zur Quest des Spielers, und somit werden auch die Emotionen des Charakters zu einem gewissen Grad übernommen.

Diese Wirkung ist jedoch nur mit sehr viel Aufwand zu erzielen. Selbst bei Spielen, die wahre Meister in der Beeinflussung von Spielern sind (wie etwa Heavy Rain), funktioniert dies nicht immer. Anders als unbewusste Ängste und Motive, die der Entwickler indirekt beeinflussen kann, unterscheiden sich das Empfinden und die Auslöser von Emotionen und Sympathien von Mensch zu Mensch. Deshalb kann kein Entwickler wirklich sicherstellen, dass die angewandten Techniken bei jedem Rezipienten auch wirklich funktionieren und zu demselben Ergebnis führen. Es gibt allerdings einige generelle Ansätze, die erfolgversprechend sind.

Empathie und Individualität

Die Identifikation mit der Spielfigur lässt sich etwa über Stereotypen und Klischees erreichen. Ethan Mars aus Heavy Rain ist beispielsweise ein absolut gewöhnlicher Familienvater. Mit dieser Rolle können sich viele Spieler identifizieren, jeder von ihnen kennt eine solche Person aus dem eigenen Umfeld, oder ist zumindest mit den gängigen Klischees vertraut. Dass dem gewöhnlichen Familienvater letztendlich Ungewöhnliches und Tragisches zustößt, ist für den Spieler folglich emotionaler, weil er sich in den Charakter hineinversetzen kann.

Die Immersion lässt sich auch durch das Anpassen der Spielfigur beeinflussen. In Mass Effect kann etwa das Aussehen von Commander Shepard selbst bestimmt werden, folglich hat der Spieler das Gefühl, einen eigenen Charakter zu spielen, obwohl sich die Geschichte selbst in einem abgesteckten Rahmen bewegt.

Ethan Mars aus **Heavy Rain** ist ein völlig durchschnittlicher Bürger. Die Spieler können sich mit einem derartigen Stereotyp einfacher identifizieren.

Die Identifikation mit der eigenen Spielfigur kann auch über eine **Individualisierung** verstärkt werden – wie zum Beispiel im Rollenspiel Mass Effect 2.

Abgesehen von der Identifikation mit den Charakteren ziehen vor allem Spiele in ihren Bann, die eine vermeintliche Entscheidungsfreiheit bieten. Der Spieler glaubt, die Handlung aktiv beeinflussen zu können und seine eigene, individuelle Geschichte zu erleben. Natürlich verhält es sich mit solchen Entscheidungen nicht anders als mit der Weggabelung aus unserem ersten Beispiel. Egal welche Entscheidung der Spieler trifft, es wird ihm nicht gelingen, eine Handlung auszuführen, deren Rahmenbedingungen der Entwickler nicht festgelegt hat.

In den letzten Jahren sind immer mehr Spiele erschienen, die sich an der unbewussten Spielerbeeinflussung probieren. Einige haben damit Erfolg und etablieren vielleicht sogar neue Methoden, andere wiederum scheitern an der Aufgabe. Doch nach wie vor kratzt die Industrie nur an der Oberfläche der Möglichkeiten, die sich mit der technischen Weiterentwicklung stetig vervielfältigen. Auch die Spieler werden sich weiterentwickeln, und somit werden die von mir erwähnten Methoden im Laufe der Zeit vermutlich obsolet werden. Umso wichtiger ist es für Entwickler, dieses Feld zu nutzen und auszuweiten, um auch zukünftigen Generationen den Spielspaß und auch die Möglichkeiten der Immersion zu bieten. Denn der Kunstgriff von heute stellt nur den Standard von morgen dar.

Richard Winterstetter

WWW
makinggames.de
- Weitere bemerkenswerte Forschungsprojekte

Applied Games in der Medizin

Gabarello: Ein Roboter als Game Controller

Die Therapiestunde wird zu einem interaktiven Erlebnis, ein Videospiel verwandelt das Therapiewerkzeug in einen Game Controller: Das interdisziplinäre Forschungsprojekt der Zürcher Hochschule der Künste offenbart Möglichkeiten und Herausforderungen für Entwickler im Feld der Applied Games.

Ein Roboter lehrt laufen: Der Physiotherapie-Roboter Lokomat® wird bei der motorischen Rehabilitation eingesetzt. Er therapiert Patienten, deren Hirnregionen, die die Steuerung des Bewegungsapparates leisten, verletzt oder gestört sind und die daher nicht mehr gehen können. Die Therapie nutzt die Eigenheit des menschlichen Gehirns, ausgefallene Funktionen durch andauerndes Training in anderen Hirnregionen erneut auszubilden. So kann die verlorene Gehfähigkeit im Laufe der Zeit wiedererlangt werden. Der Lokomat wird weltweit in Spezialkliniken eingesetzt – so auch an den Universitätskinderkliniken Zürich, deren Version speziell auf Kinder und Jugendliche zugeschnitten ist, die dort den langen Weg der Rehabilitation beschreiten.

Entscheidend für den Therapieerfolg ist intensives Training im Lokomat. Bei Kindern stellt die Motivation allerdings einen kritischen Faktor dar: Oft müssen die Therapeuten gut zureden, um sie bei Laune zu halten und ihre aktive Teilnahme an dem Bewegungsablauf zu fördern. Weil die Studienvertiefung Game Design bereits erfolgreich Serious Games für das Kinderspital entwickelt hatte, fiel die Idee zur Entwicklung eines physiotherapeutischen Videospiels für den Lokomat auf fruchtbaren Boden. Therapeuten wie Mediziner erkannten schnell das Potenzial eines exakt auf die Situation entwi-

Lokomat®

Der Lokomat ist eine angetriebene Gangorthese, die das Führen der Beine von gehbehinderten Patienten auf einem Laufband automatisiert. Das System ermöglicht eine Gangtherapie für Kinder und Erwachsene, die die Therapeuten vom anstrengenden manuellen Laufbandtraining entlastet. Zudem wird durch längere und intensivere Trainingsphasen eine effizientere Therapie ermöglicht. Das System kann mit einem Augmented Feedback Modul zur Unterstützung der Motivation des Patienten durch herausforderndes und interaktives, funktionelles Feedback in virtuellen Umgebungen verwendet werden. Mit einem zusätzlichen Lift zur Platzierung von Patienten in der Apparatur wiegt das System etwa eine Tonne, die benötigte Stellfläche umfasst fast 20 qm. Hersteller ist die Hocoma AG mit Hauptsitz in der Schweiz.

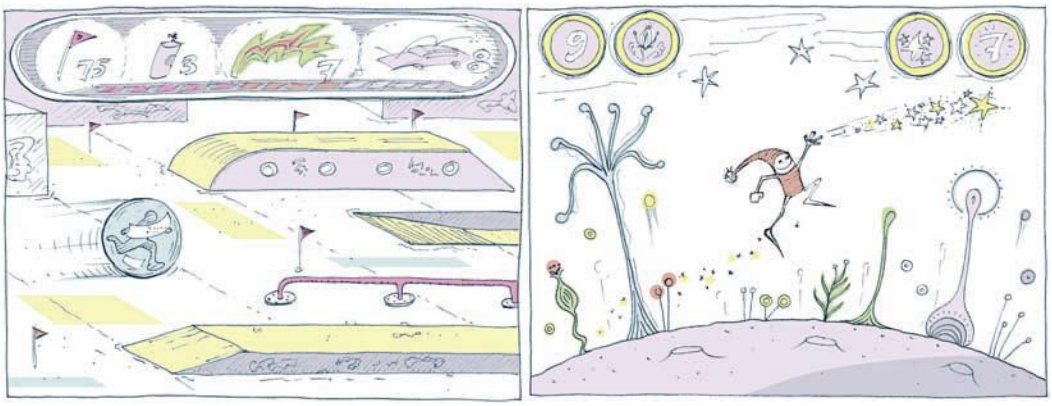


Abbildung 1: Grundlage für die spätere Umsetzung: die Spielkonzepte **Rollerball** und **Sternenstaub**.

Abbildung 2: Bereits mit diesem **frühen Prototypen** von Gabarello wurde intensiv am Roboter getestet.

Abbildung 3: Ein Screenshot der finalen Version 1.0 von **Gabarello**. Als Grafik-Engine wurde Unity3D verwendet.

ckelten »Applied Game«, das die kleinen Patienten begeistert und die Trainingssituation vergessen lässt. Daraufhin entstand ein interdisziplinäres Forschungsteam unter Beteiligung der ZHdK, des Roboterherstellers Hocoma AG, der Universitäts-Kinderkliniken Zürich sowie des Sensory Motor Systems Lab der ETH Zürich. Das Neuropsychologische Institut der Universität Zürich führte außerdem eine wissenschaftliche Evaluation des realisierten Projekts durch. Finanziert wurde das »Gabarello Vs. 1.0« (Game based rehabilitation for Lokomat) benannte Projekt von einer Stiftung zur Kinderhilfe.

Viele Spielkonzepte ...

Um im neuen Themengebiet Spielideen entwickeln zu können, bestand die erste Aufgabe des Game Design Teams in der gezielten Beobachtung von Therapiesituationen sowie in der Erhebung der Vorgaben, Bedürfnisse und Wünsche der unterschiedlichen Nutzergruppen.

Basierend auf den vorliegenden Anforderungen galt es außerdem, ein breites Spektrum an Konzepten zu entwickeln, um Abhängigkeiten und Grenzen sowie Möglichkeiten der Spielentwicklung auszuloten. Es wurden zunächst sieben Spielkonzepte entworfen (Abbildung 1), um sich dem erklärten Ziel des Spiels zu nähern – der drohenden Monotonie entgegenzuwirken und den Kindern oder Jugendlichen ein Training mit Vergnügen und ganzem Einsatz zu ermöglichen. Fünf der skizzierten Konzepte wurden anschließend in einer frühen Phase in zum Teil spielbare Prototypen (Abbildung 2) umgesetzt und vom Team im Reha-Roboter getestet. Anhand der eigenen Erlebnisse sowie des wertvollen Feedbacks von Therapeuten und Medizinern konnten die Konzepte systematisch geschärft werden. Zunehmend rückte dabei die Benutzergruppe der Physiotherapeuten in den Fokus der Entwicklung, weil diese entscheidende Hinweise über Laufrhythmen und die daraus folgende Gestaltung der Game Level einbringen konnten. Darüber hinaus wurde deutlich, dass das Spielkonzept für Patienten und Therapeuten gleichermaßen selbsterklärend sein muss und keinesfalls zu einer weiteren Beanspruchung führen darf. Denn neben dem Patienten und seinem Vorankommen im Therapiespiel muss der Therapeut zugleich die Softwareeinstellungen des Lokomat überwachen. Schließlich wurde das auf allen Ebenen aussichtsreichste Konzept realisiert, das neben attraktiv gestalteten Spielszenarien auch den Wiederspielwert über mehrere Therapiesitzungen berücksichtigt (Abbildung 3).

... ein Gameplay

Ein Ergebnis der Konzeptphase war die Anforderung nach einem stressfreien, freundlichen Szenario, das einerseits genügend Wettbewerb bietet und andererseits ausschließlich positive Anreize verwendet. Entsprechend wurde die Spielfigur gestaltet: Ein kleiner Raumfahrer wird von einer Rakete auf einem Planeten abgesetzt und spaziert auf ihm herum. Spielziel ist es, möglichst viele herumschwirrende Lichter zu entfachen, mit denen der dunkle, versteinerte Planet beleuchtet und zum Leben erweckt wird.

Das Besondere und zugleich die weitreichendste Einschränkung in der Spielsteuerung ist, dass der Spielverlauf einzig durch die Intensität der Beinbewegungen beeinflusst wird. Der Patient treibt die Spielhandlung durch das eigene vom Roboter unterstütze Gehen voran und beeinflusst so den Spielverlauf. Die Spielsteuerung werten Roboter und Spiel dabei in Echtzeit aus. Die gemessene Anstrengung eines Patienten fließt als Information direkt ins Spiel und entscheidet über die Wahl des Weges, die Fähigkeiten und den Zustand der Spielfigur. Diese kann zum Beispiel sichtlich weiter springen oder schneller laufen. Auf diese Weise werden kognitive wie koordinative Fähigkeiten gleichermaßen angesprochen (Abbildung 4).

Der Roboter lernt spielen

Die erste technische Herausforderung bestand in der Anbindung des Therapie-Roboters an das Spiel. Die drei Komponenten Lokomat mit Steuerungssoftware, Therapeuten-Interface mit Interfacesoftware und VirtualReality-Computer (im folgenden VR-Computer) mit Videospiel mussten aufeinander abgestimmt werden (Abbildung 5).

Das Kernstück in diesem Setting stellt die Steuerungssoftware des Lokomat dar, die direkt

Cornelius Müller

ist Dozent für Game Design an der ZHdK.

René Bauer

ist Dozent für Engine-Programmierung an der ZHdK.

Reto Spoerri

ist wissenschaftlicher Mitarbeiter an der ZHdK.

Prof. Ulrich Götz

Leiter der Studienvertiefung Game Design an der ZHdK.

Florian Faller, lic. phil.

unterrichtet Game Design an der ZHdK.

Abbildung 4: Gabarello und Lokomat als **Therapieeinheit** an den Universitäts-Kinderkliniken Zürich.

mit diesem verankert ist und den Roboter verwaltet. Zur Aufgabe der Steuerungssoftware gehört neben der Steuerung und Kontrolle des Automaten auch die Verarbeitung des Inputs des Therapeuten-Interfaces. Auf diesen Daten aufbauend streamt die Steuerungssoftware Livedaten über ein abgeschottetes Netzwerk. Das Videospiel auf dem VR-Computer empfängt dann diese speziell für Gabarello Vs. 1.0 zusammengestellten Rohdaten und bereitet sie auf. Als Protokoll kommt das User Datagram Protocol (UDP) zum Einsatz, das einen konstanten Stream der Live-Daten gewährleistet. Letztere werden vom Spiel als Floating-Point-Werte ausgelesen. Sie umfassen unter anderem einen Zeitindex (Timestamp), die Lage der Beine (Winkelangaben), eine Auswertung der Aktivität des Patienten (Biofeedback) sowie Spielsteuerungsinformationen (Neustart, Pause, Animationstypen etc.) der Interfacesoftware. Am Therapeuten-Interface können zusätzliche Einstellungen erfolgen, zum Beispiel Synchronisierung des Bewegungsablaufs der Spielfigur mit dem der Roboterbeine oder deren völlige Entkopplung. Das hier vorgestellte Setting mit dem Spiel Gabarello Vs. 1.0 unterliegt außerdem – weil im klinischen Rahmen angewendet – medizinischen Bestimmungen. Die Software darf zu keinem Zeitpunkt eine Gefahr für den Patienten darstellen.

Exakte Spielsteuerung

Bei Testläufen mit Prototypen im Lokomat wurde deutlich, dass das analoge Auswertungssignal über die Aktivität des Patienten starken Schwankungen unterliegt und für eine direkte Spielsteuerung nicht geeignet ist. Das gelieferte Signal erlaubt zwar Rückschlüsse auf die Aktivität der Teilnahme an der Bewegung, lässt sich aber nicht präzise genug vom Spielenden steuern. Denn in den verschiedenen Laufbewegungsphasen verfügt die Muskulatur über unterschiedlich starke Einwirkungsmöglichkeiten. Zudem sind manche der Patienten aufgrund eingeschränkter Fähigkeiten oft nicht in der Lage, die Muskulatur genauer zu kontrollieren.

Nach mehreren Trainingsexperimenten stellte sich eine Reduktion des analogen Signals auf mehrere digitale Stufen als gut steuerbar, individuell anpassbar und genügend einflussreich auf das Spielgeschehen heraus.

Spielschwierigkeit

Gabarello Vs. 1.0 beruht auf einem einfachen Spielkonzept, das wenige mentale und spielmechanische Anforderungen verlangt. Diese Designentscheidung erfolgte aufgrund der Erfahrungen, die die Spielentwickler während der Tests im Lokomat sammelten. Die andauernde Bewegung und körperliche Anstrengung durch das Training selbst bietet schon genügend Herausforderungen, das Spiel gibt dem Patienten primär einen Anreiz sowie ein Feedback über den eigenen Aktivitätsgrad. Daher wurde auf Schwierigkeitsgrade im herkömmlichen Sinn verzichtet.

Level Design

Die umgesetzte Version von Gabarello Vs. 1.0 basiert auf der Idee, dass der Spieler die Spielfigur über einen runden Planeten laufen lässt. Um eine andauernde Krümmung der Oberfläche darzustellen, müssen entweder Gravitation und Kamera andauernd angepasst werden oder – wie in diesem Fall realisiert – sich der Untergrund und nicht die Spielfigur bewegen. Diese Lösung vereinfacht auch das Leveldesign, weil so gewisse Bereiche eines Levels nicht »auf dem Kopf« modelliert werden müssen. Um dies zu vermeiden, wird der Level in Bereiche unterteilt, die nach einer definierten Reihenfolge an die jeweilige Position gesetzt werden. Ist ein Levelteil außerhalb des Sichtbereichs, kann dieser komplett aus der Szene entfernt werden.

Die Level wurden entsprechend den Anforderungen so ausgestaltet, dass sie ebenso kurzfristig wie auch längerfristig motivieren können: abwechslungsreich durch grafische Detailtiefe, aber auch herausfordernd, weil verschiedene Wege unterschiedliche Anzahlen von herumschwirrenden Lichtern erreichbar werden lassen. Geachtet wurde vor allem darauf, dass nicht nur stetig »intensives« Gehen belohnt wird, sondern auch das bewusste Abbremsen der Spielfigur. Die Langzeitmotivation wird auch dadurch unterstützt, dass der Planet so oft umgangen werden kann, bis alle Lichter entzündet sind. Dies fordert neben den motorischen auch strategische Fähigkeiten sowie ein bewusstes Auswählen der angebotenen Wege.

Jump

Weil die Spielfigur physikalischen Regeln unterworfen ist, stellte sich das Leveldesign, insbesondere die Positionierung der Absprungpunkte und die Einstellung der Absprungkraft als schwierig heraus. Für das Gestalten eines Abschnittes musste an jedem möglichen Absprungpunkt eine vertikale Kraft definiert werden, mit der die Spielfigur »hochgeschossen« wird. Zur Erleichterung des Leveldesigns wurde aufgrund dieser Erkenntnis ein Script erstellt,

Serious Game Design

Die Studienvertiefung Game Design der Zürcher Hochschule der Künste (ZHdK) ist seit 2006 im Bereich des Designs von Serious Games kontinuierlich aktiv. Zahlreiche computerbasierte Spiele, die in einem Kontext wie der medizinischen Therapie oder der Wissensvermittlung einsetzbar sind, wurden erfolgreich realisiert und befinden sich in der Anwendung. Entsprechende Kooperationsprojekte sind fester Bestandteil in Lehre und Forschung. Seit 2010 besteht außerdem der Schwerpunkt Serious Game Design, um den Anfragen aus Forschung und Wirtschaft zu Entwicklungskooperationen von Serious Games gerecht zu werden.

European Innovative Games Award 2010

Die Überraschung für das Entwicklerteam konnte nicht größer sein, denn Gabarello Vs. 1.0 wurde nicht nur für den EIGA 2010 nominiert, sondern dort auch mit dem Advancement Prize ausgezeichnet: In der Kategorie Innovative Application Methods & Environments gewann das Projekt den mit 10.000 Euro dotierten Förderpreis. Generell herausgehoben wurde in diesem Kontext das Potenzial von Videospielen, sich als Applied Games in neuen Feldern zu beweisen.

das die notwendige Kraft für einen Absprung und einen entsprechenden Zielpunkt berechnet. Für die Entwicklung einer entsprechenden Formel müssen die vorgegebenen und zu berechnenden Werte bestimmt werden.

Im Fall von Gabarello Vs. 1.0 war die Laufgeschwindigkeit gegenüber dem Boden, der Start und Zielpunkt und somit auch die Differenz zwischen diesen beiden Punkten bekannt:

```
deltaPos = sourcePos - targetPos;
```

Die Länge des Fluges wird automatisch durch die Laufgeschwindigkeit und die zurückzulegende Strecke vorgegeben:

```
flightTime = deltaPos.x / playerMovementSpeed;
```

Durch Umwandlung der Formel für die vertikale Flugbahn »y=(v*t)-((1/2)*g*t^2)« zum Beispiel mittels wolframalpha.com »solve(y=(v*t)-((1/2)*g*t^2), v)« konnte der für unsere Anwendung benötigte Algorithmus entwickelt werden. Dieser berechnet uns die benötigte Anfangsgeschwindigkeit basierend auf der Flugzeit, der Gravitation und des zu überwindenden Höhenunterschieds:

```
velocity = ((gravity * flightTime) / 2.0) + (deltaPos.y / flightTime);
```

Durch vorgängiges Verändern der Gravitation könnte zudem Einfluss auf die maximale Sprunghöhe genommen werden.

Tipps für Applied Game Design

Frühes Prototyping: Oftmals existieren bei den Kooperationspartnern oder Auftraggebern von Therapiespielen zwar erste Vorstellungen über Spielgenre und -gestaltung; konkrete Anforderungen, Einschränkungen oder Möglichkeiten können aber meist nicht genau angegeben werden. Daher empfiehlt es sich, früh Konzepte zu visualisieren und die Ideen anhand einfacher Prototypen mit allen Beteiligten zu schärfen.

Offener Austausch: Im weiteren Verlauf eines Projekts ist es ratsam, die meist gegensätzlichen Standpunkte der beteiligten Disziplinen offen abzustimmen und zu einem gemeinsamen Standpunkt der Spielentwicklung zu vereinen. Dabei geht es auch um Fragen zu den Anteilen von Spaß und Motivation gegenüber »ernsten Aspekten« in den Trainingseinheiten.

Nutzungskontext und Benutzer: Eine Begutachtung des Lehr- oder Trainingskontextes sowie der Ziel- und Peer-Gruppen (Spieler, Therapeuten, aber beispielsweise auch Eltern) ist unerlässlich und gibt Aufschluss über wichtige Details, zum Beispiel ob bestehende Phobien oder einschneidende Erlebnisse die Spielgestaltung beeinflussen und daher auf narrativer, grafischer oder interaktiver Ebene berücksichtigt werden müssen. In der Entwicklung von Gabarello Vs. 1.0 stellte sich etwa eine langsame Reaktionsgeschwindigkeit vieler Patienten als ein wichtiger Restriktionsfaktor heraus.

Spielziel: Eines sollte jedoch bei der Integration all dieser Anforderungen nicht aus den Augen verloren werden: das Ziel, ein rundum gelungenes Spiel mit einer gewissen Eigenständigkeit und einem Erlebnischarakter zu entwickeln, das den anstrengenden Prozess des Lernens unterhaltsam vergessen macht.

Passende Entwicklungsumgebung: Es empfiehlt sich, mehrere Entwicklungsumgebungen zu prüfen. Im beschriebenen Projekt wurden zwei Game Engines evaluiert, die aufgrund der geringen Kosten und der positiven Erfahrungen des Teams mit ihnen in Frage kamen: Zum einen die Panda3D Engine, die über gute Anbindungsmöglichkeiten an externe Geräte sowie eine umfangreiche Library an Zusatzfunktionen verfügt. Zum anderen kam die Unity3D Engine in Betracht, die durch ihren 3D-Editor für die Designer im Team eine erhebliche Arbeitserleichterung darstellte und darüber hinaus technisch via C# zahlreiche Bibliotheken bietet. Aufgrund der Ausrichtung des Projektes und der Zielgruppe der 6- bis 18-Jährigen wurden die visuellen Anforderungen entsprechend hoch angesetzt und daher die Unity3D Engine gewählt.

Cornelius Müller, René Bauer,
Reto Spoerri, Prof. Ulrich Götz, Florian Faller

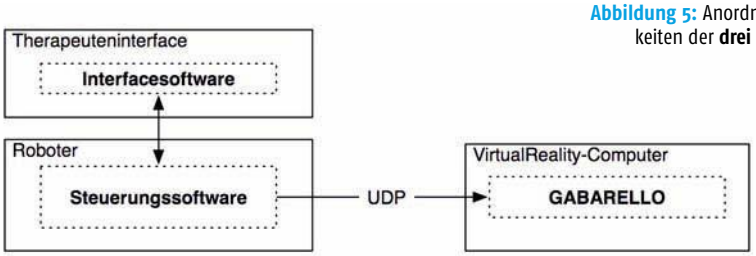


Abbildung 5: Anordnung und Abhängigkeiten der **drei Softwareeinheiten**.

Weitere Informationen

Homepage des Forschungsprojekts Gabarello:
gabarello.zhdk.ch

Homepage der Studienvertiefung Game Design an der ZHdK:
gamedesign.zhdk.ch

Homepage der Zürcher Hochschule der Künste:
zhdk.ch

Informationen zum European Innovative Games Award 2010:
innovative-games.eu

Informationen des Herstellers Hocoma AG zum Lokomat®:
hocoma.com/produkte/lokomat

WWW
makinggames.de
- Online-Praktikumsbörse mit Suchfunktion

Berufschance Spiele-Industrie
Praktikumsbörse

Sie studieren Informatik, möchten aber lieber Spiele statt Büro-Software programmieren? Sie kennen sich mit Maya aus und wollen Ihr künstlerisches Talent zum Beruf machen? Hier ist Ihre Chance: Spiele-Entwickler und Publisher aus ganz Deutschland suchen Praktikanten.

Praktikanten gesucht?

Sie suchen motivierte und talentierte Praktikanten? **info@makinggames.de, Betreff: Praktikum**
Der Abdruck ist kostenfrei.

PLZ-BEREICH 00000

online4ever Limited
Art des Praktikums: Grafik 2D/3D, Internetmarketing, Interfacedesign, Gamesupport, PHP/SQL Programmierung
Mindestdauer: 1-6 Monate
Ansprechpartner: Matthias Gruhl, info@online4ever.biz, 0351/206387
Adresse: Manitiusstraße 6, 01067 Dresden

PLZ-BEREICH 10000

Dutyfarm GmbH - Let's play together
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Programmierung (Flash, iPhone, C++), Grafik & Animation (2D/3D), Produktmanagement, QA
Mindestdauer: 6 Monate
Ansprechpartner: Fabian Masche, jobs@dutyfarm.de, 030 / 30368430
Adresse: Oranienstraße 6, 10997 Berlin

Exozet Games
Art des Praktikums: Nach Absprache
Bereich: Maya 3D Max Modelling, Animation
Mindestdauer: 6 Monate
Ansprechpartner: Bartol Ruzic, bewerbung@exozet.com
Adresse: Rotherstraße 20, 10245 Berlin

Future Interactive
Art des Praktikums: Studienbegleitend
Bereich: Java-Programmierung, C++-Programmierung
Mindestdauer: Nach Absprache
Ansprechpartner: Felix Kollmann, fkollmann@futureint.de, 030 / 810 593 36
Adresse: Gartenstraße 1, 14169 Berlin

GameDuell
Art des Praktikums: Studienbegleitend, nach Absprache
Bereich: Programmierung, Marketing, Controlling/Finanzen
Mindestdauer: 6 Monate
Ansprechpartner: Tina Haile, jobs@gameduell.de, 030 / 288 768 211
Adresse: Ackerstraße 14-15, 10115 Berlin

Games Academy
Art des Praktikums: Vollzeit
Bereich: GAMES Marketing/PR, Print, Ambient, Online, Social Communities, Web
Mindestdauer: 6 Monate
Ansprechpartner: Tina Pietsch, tina.pietsch@games-academy.de, Onlinebewerbung erwünscht, 030/29779120
Adresse: Rungestraße 20, 10179 Berlin

K1010 Media
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Spiele-Entwicklung (Flash), PHP-Programmierer, Webdesign, Marketing
Mindestdauer: 3 Monate
Ansprechpartner: Martina Misikova, bewerbung@k1010.de, 030 / 57709936
Adresse: Erkelenzdamm 59/61, 10999 Berlin

MYTIVI - aktiv fernsehen
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Entwicklung und Konzeption TV-Games (Transaktions-TV)
Mindestdauer: 3 Monate
Ansprechpartner: Daniel Schmitt, jobs@mytivi.com
Adresse: Oranienstraße 6, 10997 Berlin

wooga – world of gaming
Art des Praktikums: Vollzeit
Bereich: Quality Assurance/ Testen neuer Spielversionen
Mindestdauer: 3 Monate
Ansprechpartner: Peggy Beschnitt, peggy@wooga.com
Adresse: Saarbrücker Str. 38, 10405 Berlin

Yager Development
Art des Praktikums: Vollzeit, Praktikumssemester, studienbegleitend, praktische Diplomarbeit
Bereich: Programmierung (AI, 3D, Physik), Grafik (2D/3D), Scripting, Animation, Leveldesign
Mindestdauer: 3 Monate
Ansprechpartner: Eike Stolley, eike.stolley@yager.de
Adresse: Pfuelstraße 5, 10997 Berlin

PLZ-BEREICH 20000

Bigpoint
Art des Praktikums: Vollzeit
Bereich: Produktanalyse/Produktmanagement, Gamedesign
Mindestdauer: 3 Monate
Ansprechpartner: Human Resources, personal@bigpoint.net, Betreff bitte »Praktikumsbörse, Produktanalyse« oder »Praktikumsbörse, Gamedesign«
Adresse: Christoph-Probst-Weg 3, 20251 Hamburg

Borgmeier Media Communication
Art des Praktikums: Praktikum für Studenten mit Schwerpunkt Marketing
Bereich: Games-PR, Corporate-PR, Marketing
Mindestdauer: 3 Monate
Ansprechpartner: Tanja Deilecke, deilecke@agentur-borgmeier.de, 040 / 413 09 60
Adresse: Rothenbaum Chaussee 5, 20148 Hamburg

CNT Media
Art des Praktikums: Nach Absprache (Übernahme möglich)
Bereich: Kommunikation, Level-Design, Projekt-Planung
Mindestdauer: 3 Monate
Ansprechpartner: Cornel Hillmann, ch@cnt-media.net
Adresse: Süderstraße 159a, 20537 Hamburg

Daedalic Entertainment
Art des Praktikums: Vollzeit
Bereich: Programmierung, Grafik, Game Design, Marketing,
Mindestdauer: 3 Monate, besser 6 Monate
Ansprechpartner: Dominik Heineccius, dm@daedalic.de, 040/ 432 612 72
Adresse: Papenreye 53, Workport Unit 2, 22453 Hamburg

dtp entertainment
Art des Praktikums: Nach Absprache
Bereich: PR, Marketing, Grafik, Test, Produktmanagement
Mindestdauer: 6 Monate
Ansprechpartner: Claas Wolter, c.wolter@dtp-ag.com, 040 / 669 910 43
Adresse: Goldbekplatz 3-5, 22303 Hamburg

Goodgame Studios
Art des Praktikums: nach Absprache
Bereich: Online Marketing, Web-Promotion, Game Development
Mindestdauer: 3 Monate
Ansprechpartner: Patrick Abrar, jobs@goodgamestudios.com
Adresse: Theodorstraße 42-90, Haus 4a, 22761 Hamburg

Indigo Pearl
Art des Praktikums: Nach Absprache
Bereich: PR-Agentur
Mindestdauer: 6 Monate
Ansprechpartner: Torsten Oppermann, to@indigopearl.de, 040 / 253 302 67
Adresse: Richardstraße 45, 22081 Hamburg

InnoGames
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Online-Marketing/Mediadesign, Grafikdesign, Systemadministration, Softwareentwicklung
Mindestdauer: 4 Monate
Ansprechpartner: Anna Sparke, career@innogames.de
Adresse: Harburger Schloßstraße 28, 21079 Hamburg

INPROMO
Art des Praktikums: Vollzeit
Bereich: Social Media Marketing für Entertainment-Kunden
Mindestdauer: 6 Monate
Ansprechpartner: Kerstin Saathoff, bewerbung@inpromo.de
Adresse: Banksstrasse 2b, 20097 Hamburg

northworks
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Online-Marketing/Mediadesign, Grafikdesign, Systemadministration, Softwareentwicklung
Mindestdauer: 4 Monate
Ansprechpartner: Anna-Katja Sparke
Adresse: Stahltwiete 23, 22761 Hamburg

Splitscreen Studios
Art des Praktikums: Vollzeit
Bereich: Community Management, Programmierung, 3D-Grafik
Mindestdauer: 4 Monate
Ansprechpartner: Gustaf Stechmann, gus@splitscreenstudios.com
Adresse: Neuer Kamp 30, 20357 Hamburg

Spotsonfire
Art des Praktikums: Nach Absprache
Bereich: Marketing, Programmierung, 2D-Design, Illustration, Animation
Mindestdauer: 3 Monate
Ansprechpartner: jobs@spotsonfire.com, 040 / 209 346 330
Adresse: Hongkongstraße 7, 20457 Hamburg

Ticking Bomb Games
Art des Praktikums: Vollzeit
Bereich: Programmierung, Grafik
Mindestdauer: 3 bis 6 Monate
Ansprechpartner: Marco Schultz, jobs@ticking-bomb-games.de
Adresse: Böckmannstraße 15, 20099 Hamburg

wwg worldwidegames / Playnik
Art des Praktikums: Vollzeit
Bereich: Game Design, Product Management, Business Development, Community Management
Mindestdauer: 3 Monate
Ansprechpartner: Susan Kohnke, sk@playnik.com
Adresse: Richardstraße 45, 22081 Hamburg

PLZ-BEREICH 30000

Reakktor Media
Art des Praktikums: Nach Absprache
Bereich: 3D-Grafik, Web-Programmierung, Leveldesign
Mindestdauer: 6 Monate
Ansprechpartner: Kirk Lenke, linke@reakktor.com, 0511 / 540 980
Adresse: Arnswaldtstraße 10, 30159 Hannover

Rocketbox Studios
Art des Praktikums: Nach Absprache
Bereich: 3D-Grafik
Mindestdauer: 5 Monate
Ansprechpartner: Markus Wojcik, jobs@rocketbox.de, 0511 / 935 72 80
Adresse: Vahrenwalder Straße 7, 30165 Hannover

Xybris Interactive
Art des Praktikums: Nach Absprache
Bereich: Programmierung (Java, PHP), Grafik, Webdesign (CBS)
Mindestdauer: 3 Monate
Ansprechpartner: Christian Sauer, info@xybris.com
Adresse: Apfelstraße 12a, 33613 Bielefeld

PLZ-BEREICH 40000

bhv Software
Art des Praktikums: Studienbegleitend, Vollzeit
Bereich: Produktmanagement, Games, Non-Games, Edutainment, Herstellung, PR
Mindestdauer: 3 Monate
Ansprechpartner: Dieter Dedeke, bewerbung@bhv.de, 02131 / 765 01,
Adresse: Novesiastraße 60, 41564 Kaarst

Ubisoft / Blue Byte
Art des Praktikums: Nach Absprache
Bereich: Programmierung, Game Design, Level Design, Grafik, Projektmanagement
Ansprechpartner: siehe unter http//job.ubisoft.com/MAIN/careerportal, 0211 / 338 000
Adresse: Adlerstraße 74, 40211 Düsseldorf

Zone 2 media
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Programmierung, 3D-Grafik
Mindestdauer: 6 Monate, nach Absprache
Ansprechpartner: Holger Offermanns, h.offermanns@zone2.de, 02150 / 705660
Adresse: Am Wasserturm 6, 40668 Meerbusch

PLZ-BEREICH 50000

Bright Future
Art des Praktikums: Vollzeit, Praxis-Semester
Bereich: C++-Programmierung
Dauer: 4 Monate, besser 6 Monate
Ansprechpartner: Gerald Köhler, jobs@brightfuture.de, 0221 / 78982171
Adresse: Im Zollhafen 15-17, 50678 Köln

EA
Art des Praktikums: Vollzeit
Bereich: PR, Marketing, Sales, Verwaltung
Mindestdauer: 4 Monate
Ansprechpartner: Volker Pinsdorf, www.electronic-arts.de, 0221 / 975820
Adresse: Im Zollhafen 15-17, 50678 Köln

EA Phenomic
Art des Praktikums: Vollzeit
Bereich: 2D/3D-Grafik, 2D-Grafik, 3D-Grafik, Programmierung, IT-Administration, Qualitätssicherung
Mindestdauer: 4 Monate
Ansprechpartner: Volker Pinsdorf, www.phenomic.de, 06132 / 78350
Adresse: Binger Straße 38, 55218 Ingelheim

Egosoft
Art des Praktikums: Studienbegleitend oder normal
Bereich: 3D-Grafik, Design, Programmierung
Dauer: Mindestens 6 Monate
Ansprechpartner: Bernd Lehahn, bernd@egosoft.com
Adresse: Heidestraße 4, 52146 Würselen

Headup Games
Art des Praktikums: Vollzeit
Bereich: PR, Marketing, und Producing
Dauer: 6 Monate
Ansprechpartner: Michael Zolna, info@headupgames.com
Adresse: Nordstraße 104, 52353 Düren

Independent Arts Software
Art des Praktikums: Nach Absprache
Bereich: 3D-Grafik und Animation, vorzugsweise mit Softimage XSI, 2D-Grafik und Animation für GBA-Projekte, Tool, 3D-Programmierung unter C++
Mindestdauer: 3 Monate, besser 6 Monate
Ansprechpartner: Holger Kuchling, ceo@independent-arts-software.de, 02381 / 688 292
Adresse: Münsterstraße 5, 59065 Hamm

Related Designs
Art des Praktikums: Studienbegleitend, Vollzeit, Praktikumssemester
Bereich: Programmierung, Grafik, Game Design
Mindestdauer: 6 Monate
Ansprechpartner: Svenja Büll, jobs@related-designs.de
Adresse: Römerpassage 1, 55116 Mainz

Spacetale
Art des Praktikums: Studienbegleitend, Vollzeit
Bereich: 2D/3D-Grafik
Mindestdauer: 3 Monate
Ansprechpartner: Michael Wolf, jobs@spacetale.de
Adresse: Boppstraße 26, 55118 Mainz

PLZ-BEREICH 60000

bitComposer Games
Art des Praktikums: Nach Absprache
Bereich: Marketing & PR
Mindestdauer: 3 Monate
Ansprechpartner: Lukasz Szczepanczyk, info@bit-Composer.com, 06196 / 7793810
Adresse: Mergenthalerallee 79-81, 65760 Eschborn

Cooee GmbH
Art des Praktikums: Nach Absprache
Bereich: QA, Grafik, Webprogrammierung
Mindestdauer: 3 Monate
Ansprechpartner: Boris Karklins, jobs@clubcooee.com, 06301 / 703 290
Adresse: Opelstraße 10, 67661 Kaiserslautern

Deck 13
Art des Praktikums: Nach Absprache
Bereich: 2D/3D-Design, 3D Character Artist/Animator, Concept Art, PR-Management, Webdesign
Mindestdauer: 6 Monate
Ansprechpartner: Sarah Wrensch, jobs@deck13.de
Adresse: Baselerstraße 46, 60329 Frankfurt

Dutyfarm GmbH
Art des Praktikums: Studienbegleitend, Praxis-Semester
Bereich: Programmierung (Flash Actionscript, PHP, JSP, Datenbankprogrammierung), Grafik & Gamedesign (2D/3D), Produktmanagement, PR, QA
Mindestdauer: 6 Monate
Ansprechpartner: Daniel Schmitt, jobs@dutyfarm.com
Adresse: Oranienstraße 6, 10997 Berlin

extra toxic
Art des Praktikums: Studienbegleitend, nach Absprache
Bereich: Programmierung
Mindestdauer: 3 Monate
Ansprechpartner: Ivica Aracic, jobs@extratoxic.com, 069/ 40897143
Adresse: Hanauer Landstraße 196a, 60314 Frankfurt

Keyfactor
Art des Praktikums: Nach Absprache
Bereich: Produktionsassistenz, Marketing & PR, Asset & Content Management
Mindestdauer: 6 Monate
Ansprechpartner: Frank Fay, jobs@key-factor.de
Adresse: Rudolf-Diesel-Straße 20a, 65760 Eschborn/Ts.

World Cyber Games
Art des Praktikums: Nach Absprache
Bereich: PR
Mindestdauer: 3 Monate
Ansprechpartner: Thomas von Treichel, tvt@ac-entertainment.com, 06152 / 985482
Adresse: Im Neugrund 6-13, 64521 Groß Gerau

PLZ-BEREICH 70000

Feenix
Art des Praktikums: Vollzeit, Praxis-Semester
Bereich: Programmierung (Java, PHP)
Mindestdauer: 6 Monate
Ansprechpartner: Uwe Spitzmüller, uwe.spitzmueller@feenix.de
Adresse: Durmersheimerstraße 55, 76185 Karlsruhe

Filmakademie Baden-Württemberg
Art des Praktikums: Studienbegleitend, Vollzeit
Bereich: Character Design, Engine Entwicklung, Shader Development, Cell BE Programmierung, Maya Api Tool Entwicklung
Mindestdauer: 6 Monate
Ansprechpartner: Volker Helzle, research@filmakademie.de
Adresse: Mathildenstraße 20, 71638 Ludwigsburg

Gameforge
Art des Praktikums: Studienbegleitend, Vollzeit
Bereich: Grafik-Design, Web-Design, Programmierung PHP, 3D-Modelling, Game Design, Community Management
Mindestdauer: 3 bis 6 Monate
Ansprechpartner: Anett Graf, jobs@gameforge.de
Adresse: Albert-Nestler-Straße 8, 76131 Karlsruhe

Korion Simulation
Art des Praktikums: studienbegleitend, Praxis-Semester, Diplom, Bachelor
Bereich: Flash-Entwicklung, AS3-Programmierung, Webdesign, 3ds Max
Mindestdauer: 3 Monate
Ansprechpartner: Oliver Korn, oliver.korn@korion.de
Adresse: Mömpelgardstraße 16, 71640 Ludwigsburg

kr3m. media
Art des Praktikums: Nach Absprache
Bereich: Illustration, Gestaltung, Konzeption von Online-Spielen, Webdesign, Projektmanagement, Programmierung von Flash-Games, PHP/HTML/XML
Mindestdauer: mindestens 3 Monate
Ansprechpartner: Patricia Kuban, patricia.kuban@kr3m.com, 0175 / 5635164
Adresse: Kaiserstraße 158, 76133 Karlsruhe

SEW-EURODRIVE
Art des Praktikums: Vollzeit
Bereich: Managed DirectX Programmierung
Mindestdauer: 3 Monate
Ansprechpartner: Theres Petrasch, theres.petrasch@sew-eurodrive.de, 07251 / 751908
Adresse: Ernst-Blickle-Straße 42, 76646 Bruchsal

Sixteen Tons Entertainment
Art des Praktikums: Vollzeit, Studienbegleitend
Bereich: 2D/3D-Grafik, C++ Programmierung, Leveldesign
Mindestdauer: 6 Monate
Ansprechpartner: Lars Klinksiek, jobs@sixteen-tons.de
Adresse: Karlstraße 3, 72072 Tübingen

Trinigy
Art des Praktikums: Vollzeit, Praxis-Semester
Bereich: Softwareentwicklung im 3D-Game-Engine-Umfeld
Mindestdauer: 3 Monate
Ansprechpartner: Felix Röken, felix.roeken@trinigy.de, 07121 / 986993
Adresse: INKA-Businesspark, Arbachtalstr. 6, 72800 Eningen

PLZ-BEREICH 80000

B-Alive
Art des Praktikums: Nach Absprache
Bereich: Datenmanagement, Gamedesign und assistierende Projektleitung, 3D-Grafik, Software-Entwicklung C++
Mindestdauer: 6 Monate
Ansprechpartner: Arndt Schlichtig, jobs@b-alive.de
Adresse: Biberacher Straße 38, 88444 Ummendorf

CreateCtrl
Art des Praktikums: Vollzeit oder Teilzeit, Praxissemester, Übernahme möglich
Bereich: Entwicklung von Multiplayer-Spielen mit Flash, Flex und C# sowie Portalentwicklung mit JBuilder
Mindestdauer: 1 bis 6 Monate
Ansprechpartner: Klaus Herrmann, klaus.herrmann@createctrl.com, 089 / 55277736
Adresse: Bayerstraße 85a, 80335 München

Disney Interactive
Art des Praktikums: Vollzeit
Bereich: Marketing
Mindestdauer: 6 Monate
Ansprechpartner: Christina Ettelbrück, christina.ettelbrueck@disney.com
Adresse: Kronstadter Straße 9, 81677 München

Marchsreiter Communications
Art des Praktikums: Vollzeit
Bereich: PR
Mindestdauer: 3 Monate
Ansprechpartner: Dieter Marchsreiter, dm@marchsreiter.com
Adresse: Guldeinstraße 41a, 80339 München

Realmforge Studios
Art des Praktikums: Nach Absprache
Bereich: Programmierung
Mindestdauer: 6 Monate
Ansprechpartner: Christian Wolfertstetter, christian.wolfertstetter@realmforgestudios.com
Adresse: Hofer Straße 15, 81737 München

Remote Control Productions
Art des Praktikums: Vollzeit
Bereich: Game Design, Produktmanagement, Projektmanagement, Producing
Mindestdauer: 6 Monate
Ansprechpartner: Johannes Lesser, jobs@r-control.de
Adresse: Sendlinger-Tor-Platz 6, 80336 München

SevenOne Intermedia
Art des Praktikums: Vollzeit
Bereich: Marketing, Produkt Management, Game-Producing
Mindestdauer: 6 Monate
Ansprechpartner: www.prosiebensat1-jobs.de
Adresse: Medienallee 6, 85774 Unterföhring

Take 2 Interactive
Art des Praktikums: Vollzeit
Bereich: Marketing, Produktmanagement
Mindestdauer: 6 Monate
Ansprechpartner: Jochen Till, jochen.till@take2.de, 089 / 27822100
Adresse: Agnesstraße 14, 80798 München

Travian Games
Art des Praktikums: Games Analyse & Evaluation
Bereich: Publishing
Mindestdauer: 6 Monate
Ansprechpartner: Christiane Nißl, jobs@traviangames.com
Adresse: Wilhelm-Wagenfeld-Straße 22, 80807 München

PLZ-BEREICH 90000

HandyGames
Art des Praktikums: Studienbegleitend, Vollzeit
Bereich: Grafik, Programmierung, Game- & Leveldesign
Mindestdauer: 6 Monate
Ansprechpartner: Markus Kassulke, jobs@handy-games.com, 09334 / 97570
Adresse: i_Park Klingholz 13, 97232 Giebelstadt

Pixel Natives
Art des Praktikums: Nach Absprache
Bereich: Programmierung(C#, Java),Webprogrammierung
Mindestdauer: 3 Monate, besser 6 Monate
Ansprechpartner: Wolfgang Kurz, w.kurz@buiss-ultimo.de
Adresse: Clermont-Ferrand-Allee 36, 93049 Regensburg

qforge
Art des Praktikums: Vollzeit
Bereich: Konzeption, Programmierung (Flash/Actionscript), Systemintegration/Servermanagement (Debian Linux)
Mindestdauer: 3 Monate
Ansprechpartner: Martin Schiele, M.Schiele@qforge.de
Adresse: Südwestpark 37-41, 90449 Nürnberg

Upjers
Art des Praktikums: Vollzeit, nach Absprache
Bereich: 3D-Grafik, Programmierung, PR
Mindestdauer: 6 Wochen
Ansprechpartner: Marika Schmitt, m.schmitt@upjers.com
Adresse: Seehofstraße 46, 96117 Memmelsdorf

A-1000

Sproing
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Quality Assurance, Production
Mindestdauer: 5 Monate
Ansprechpartner: Johanna Schober, jobs@sproing.com, +43 / (0)1 / 604302838
Adresse: Fernkorngasse 10, 1100 Wien, Österreich

JEDE AUSGABE MIT FIRMENREGISTER UND PRAKTIKUMSBÖRSE
making games MAGAZIN
MAGAZIN FÜR SPIELE-ENTWICKLUNG UND BUSINESS-DEVELOPMENT
01/2011 € 6,90
2011
TECHNOLOGIE FÜR 2011
S3D, KINECT, CROSS-PLATFORM: WIE CRYTEK, RARE, UBISOFT MONTREAL SCHON HEUTE DIE TECHNIK VON MORGEN
GAME DESIGN
WIE MAN DAS BALANCING BEREITS IN DER KONZEPTPHASE ERLEDIGT
EXKLUSIV-INTERVIEW
DIE PROJEKTE UND PERSPEKTIVEN DES DEUTSCHEN ZYNGA-STUDIOS
bis zu
51%
sparen!

SONDER-RABATTE für Unternehmen der Spielebranche!

	Ersparnis
Standard-Jahresabo (für jeden!) (Nur noch 28,98 Euro / Abo)	**-30%**
5-9 Jahresabos (Firmen, Schulen etc.) (Nur 26,95 Euro / Abo)	**-35%**
10-19 Jahresabos (Nur 24,95 Euro / Abo)	**-40%**
Mehr als 20 Jahresabos (Nur 19,95 Euro / Abo)	**-51%**

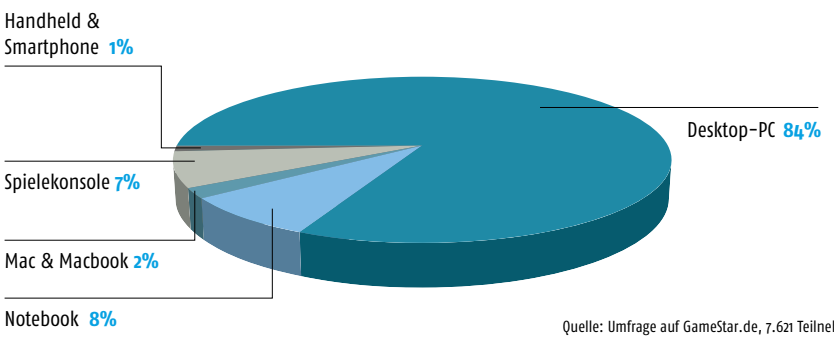
Profitieren Sie von den günstigen Firmenangeboten und bestellen Sie online oder telefonisch:

Infos & Bestellung: **www.makinggames.de/firmenabo**
oder **Tel.: 0180/5 72 72 52-275*** Mo-Fr 8-20 Uhr (*aus dem dt. Festnetz nur 0,14 € pro Min., Mobilfunkpreise max. 0,42 € pro Min.)

Ausgewählte Umfragen
Dataflow

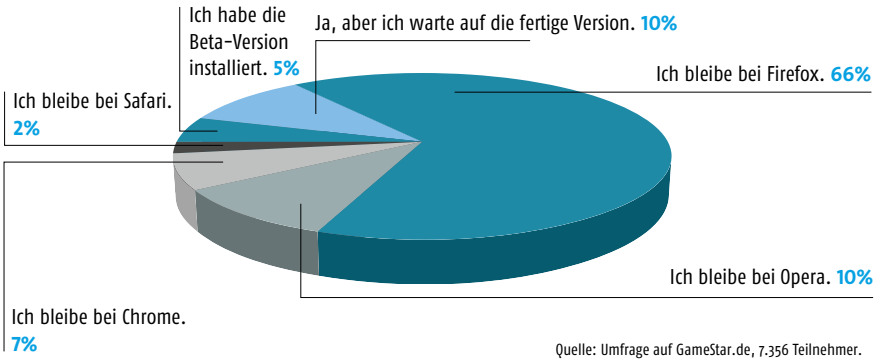
Die Redaktionen von GamePro und GameStar machen monatlich zahlreiche Umfragen unter Heftkäufern und Homepage-Usern. Eine kleine Auswahl finden Sie auf dieser Seite.

Auf welcher Plattform spielen Sie vorrangig?



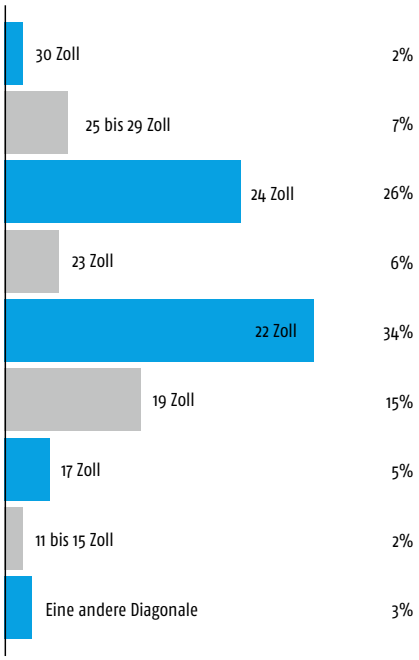
Quelle: Umfrage auf GameStar.de, 7.621 Teilnehmer

Interessieren Sie sich für den Internet Explorer 9?


Ich habe die Beta-Version installiert. 5%
Ja, aber ich warte auf die fertige Version. 10%
Ich bleibe bei Firefox. 66%
Ich bleibe bei Safari. 2%
Ich bleibe bei Opera. 10%
Ich bleibe bei Chrome. 7%

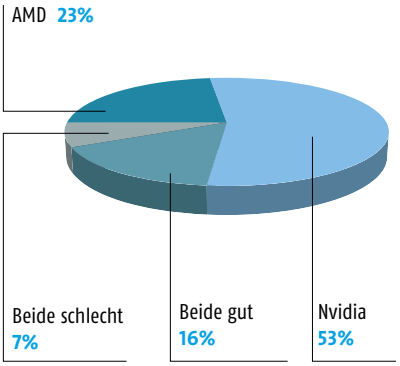
Quelle: Umfrage auf GameStar.de, 7.356 Teilnehmer.

Welche Bildschirm-Diagonale hat Ihr Monitor?



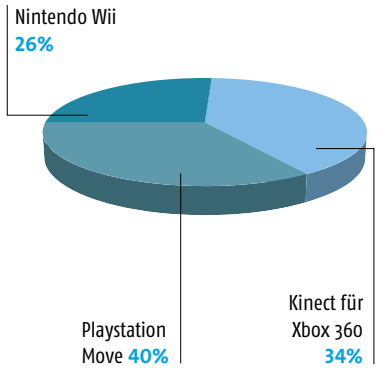
Quelle: Umfrage auf GameStar.de, 6.363 Teilnehmer

Wer hat die besseren Grafikkarten-Treiber?



Quelle: Umfrage auf GameStar.de, 5.833 Teilnehmer.

Sony, Microsoft und Nintendo haben nun alle drei eine Bewegungssteuerung am Markt. Welche ist die beste?



Quelle: Umfrage auf GamePro.de, 4.980 Teilnehmer

WWW
makinggames.de

- Online-Register mit Suchfunktion. Zu jedem Key Player finden Sie zusätzlich ein detailliertes Firmenportrait.

Helfen Sie uns!

Sie haben einen Fehler entdeckt oder vermissen Ihre Firma? Dann schreiben Sie uns: **info@makinggames.de, Betreff: Firmenregister**

Die Branche im Überblick
Firmenregister

Vergessen Sie die Auskunft! Hier finden Sie sämtliche uns bekannten Publisher, Entwickler, Dienstleister, Ausbilder und Verbände der deutschsprachigen Spiele-Industrie – von Acony bis Zuxxez.

PUBLISHING/DISTRIBUTION

Firma	Straße/Hausnummer	Postleitzahl/Ort		Telefon	E-Mail
505 Games	Brunnfeld 2-6	93133	Burglengenfeld	09471 / 3088792	
Activision Blizzard Deutschland	Fraunhoferstraße 7	85737	Ismaning	089 / 9998170	info@activision.de
Aerosoft	Lindberghring 12	33142	Büren	02955 / 760310	info@aerosoft.de
Altigi	Theodorstr. 42-90, Haus 4a	22761	Hamburg	040 / 4085399366	info@altigi.de
Application Systems Heidelberg	Pleikartsförsterhof 4/1	69124	Heidelberg	06221 / 300002	kontakt@application-systems.de
BHV	Novesiastraße 60	41564	Kaarst	02131 / 76501	info@bhv.de
BIGPOINT	Christoph-Probst-Weg 3	20251	Hamburg	040 / 8814130	info@bigpoint.net
bitcomposer Games	Mergenthalerallee 79-81	65760	Eschborn	06196 / 7793810	info@bit-composer.com
Burda:ic	Arabellastraße 23	81925	München		info@burda-ic.com
Capcom - CEG Interactive Entertainment	Barmbeker Straße 4b	22303	Hamburg	040 / 6965620	info@capcomeuro.de
CDV	Hanauer Landstraße 161-173	60314	Frankfurt am Main	069 / 40563730	mail@cdv.de
Comport Interact!ve	Eberleinstraße 46	65195	Wiesbaden	0611 / 53169783	info@comport-interactive.com
Daedalic Entertainment	Papenreye 53-Workport Unit 2	22453	Hamburg	040 / 432612720	info@daedalic.de
Dartmoorsoft	Postfach 6161	32647	Lemgo	05261 / 660731	info@dartmoorsoft.com
Disney Interactive Studios	Kronstadter Straße 9	81677	München	089 / 993400	info@disney.com
dtp entertainment	Goldbekplatz 3-5	22303	Hamburg	040 / 6699100	info@dtp-ag.com
dtp young entertainment	Goldbekplatz 3-5	22303	Hamburg	040 / 6699100	info@dtp-young.com
Electronic Arts	Im Zollhafen 15-17	50678	Köln	0221 / 975820	
Flashpoint	Valvo Park - Haus 5c / Tarpen 40	22419	Hamburg	040 / 710060	info@flashpoint.de
Frogster Interactive Pictures	Hardenbergstraße 9A	10623	Berlin	030 / 28470110	info@frogster.de
GameDuell	Greifswalder Straße 212-213	10405	Berlin	030 / 288768211	impressum@gameduell.de
Gameforge	Albert-Nestler-Straße 8	76131	Karlsruhe	0721 / 3548080	info@gameforge.de
Gameloft	Charlottenstraße 68	10117	Berlin	030 / 20188429	gregory.wintgens@gameloft.com
Gamesload	T-Online-Allee 1	64295	Darmstadt	06151 / 6800	presse@gamesload.de
GameTwist	Mariahilfer Straße 47/1/102	1060	Wien	+43 / 1494 / 5056	3rdparty@gametwist.com
gamigo	Behringstraße 16b	22765	Hamburg	040 / 226305250	gamigoag@gamigo-mail.de
Goodgame Studios	Theodorstraße 42-90, Haus 4a	22761	Hamburg	040 / 85192490	info@goodgamestudios.com
Halycon Media	An der Autobahn 24	23858	Reinfeld	04533 / 61090	infomail@halycon.de
Headup Games	Nordstraße 104	52353	Düren	02421 / 4868700	info@headupgames.com
Intenium	Neuer Pferdemarkt 1	20359	Hamburg	040 / 25494296	info@intenium.de
JoWood	Heiligenstädterstraße 201-203	A-1190	Wien	+43 / 1 / 379090	office@jowood.com
Kalypso Media	Prinz-Carl-Anlage 36	67547	Worms	06241 / 502240	info@kalypsomedia.com
Koch Media	Lochhamer Straße 9	82152	Planegg/München	089 / 24245120	office@kochmedia.de
Konami	Berner Straße 103-105	60437	Frankfurt am Main	069 / 9855730	support@konami.de
Mail.ru	Ankelmannsplatz 1	20537	Hamburg	040 / 4143420	contact_de@corp.mail.ru
Microsoft	Konrad-Zuse-Straße 1	85716	Unterschleißheim	089 / 31760	kunden@microsoft.com
Millform	Ringstraße 110	CH-4106	Therwil	+41 / 61 / 7220712	werner.sala@millform.ch
Morphicon	Lindwurmstraße 88	80337	München	089 / 45235460	info@morphicon.de
MultiMediaManufaktur	Ludwigstraße 33-37	60327	Frankfurt am Main	069 / 974600	info@multimediamanufaktur.com
Namco Bandai Partners	Carl-Benz-Straße 21	60386	Frankfurt am Main	069 / 24449100	de.info@namcobandaipartners.com
NBG	Brunnfeld 2-6	93133	Burglengenfeld	09471 / 70170	info@nbg-online.de
Nintendo	Nintendo Center	63760	Grossostheim	06026 / 9500	info@nintendo.de
Nokia	Balcke-Dürr-Allee 2	40882	Ratingen	02102 / 89280	
Novitas	Frankfurter Straße 2	38122	Braunschweig	02574 / 9270	kontakt@greenpepper.de
Phenomedia	Friedrich-Ebert-Straße 14	44866	Bochum	02327 / 9970	info@phenomedia.com
qforge	Südwestpark 37 - 41	90449	Nürnberg	0911 / 3092097	info@qforge.de
Ravensburger Digital	Agnes-Pockels-Bogen 1	80992	München	089 / 12228180	tellme@ravensburger-digital.com
redspotgames	Postfach 310464	80104	München	089 / 4036370	info@redspotgames.com
Rondomedia	Limitenstraße 64-78	41236	Mönchengladbach	02166 / 618660	info@rondomedia.de
RTL Games	Achnerstraße 1036	50858	Köln	0221 / 4560	entwickler@rtl-interactive.de
S.A.D.	Rötelbachstraße 91	89079	Ulm	07305 / 96290	kontakt@s-a-d.de
Sega	Rablstraße 24	81669	München	089 / 22848000	info@sega.de
SevenOne Intermedia	Medienallee 6	85774	Unterföhring	089 / 950710	info@sevenonegames.de
Sony Computer Entertainment	Frankfurter Straße 233	63263	Neu-Isenburg	06102 / 4330	scee_presscentre@scee.net
Square Enix	Domstraße 17	20095	Hamburg	040 / 30633400	
Take 2	Agnesstraße 14	80798	München	089 / 278220	info@take2.de
THQ	Kimplerstraße 278	47807	Krefeld	02151 / 41890	thq@thq.de
Tipp24 Entertainment	Straßenbahnring 11	20251	Hamburg		info@t24e.de
Tivola Electronic Publishing	Oeverseestraße 10-12	22769	Hamburg	030 / 5363580	mail@tivola.de
Travian Games	Wilhelm-Wagenfeld-Straße 22	80807	München	089 / 324915104	mail@traviangames.com
Ubisoft	Adlerstraße 74	40211	Düsseldorf	0211 / 338000	info@ubisoft.de
Zuxxez	Rittnertstraße 36	76227	Karlsruhe	0721 / 464720	info@zuxxez.com

DEVELOPMENT

Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
3d-io	Rheingaustraße 94	65203 Wiesbaden	0611 / 5829602	info@3d-io.com
49Games	Waterloohain 9	22769 Hamburg		kontakt@49games.de
Acony	Mönchweilerstraße 1/8	78048 Villingen-Schwenn.	07721 / 9983829	info@aconygames.com
Aitainment	Konsul-Smidt-Straße 8f	28217 Bremen	0421 / 6195640	info@aitainment.de
Adrastea	Bielefelder Str. 5	90425 Nürnberg	0700 / 42300000	info@adrastea.com
Altigi	Theodorstr. 42-90, Haus 4a	22761 Hamburg	040 / 4085399366	info@altigi.de
B-Alive	Biberacher Straße 38	88444 Ummendorf	07351 / 441307	admin@b-alive.de
BigBlackBlock Gamestudio	Am Veenteich 1	46147 Oberhausen	0208 / 6101995	info@bigblackblock.com
BIGPOINT	Christoph-Probst-Weg 3	20251 Hamburg	040 / 8814130	info@bigpoint.net
binary madness	Bundesallee 55	10715 Berlin	030 / 25093746	info@binarymadness.de
Bit Barons c/o Mediadesign Hochschule	Claudius-Keller-Straße 7	81669 München	089 / 41617744	info@bitbarons.com
Bitfield	Rigaer Straße 56	10247 Berlin	030 / 400451 81	kontakt@bitfield.de
Blue Byte	Adlerstraße 74	40211 Düsseldorf	0211 / 338000	info@bluebyte.de
BlurredVision	Glashütter Weg 80	22844 Norderstedt	040 / 53532683	info@blurredvision.de
Bronx Studios Entertainment	Lyoner Straße 14	60528 Frankfurt	069 / 66554486	info@bronxstudios.com
BSS web consulting	Dinglerstraße 5	45145 Essen	0201 / 5200044	info@bss-consulting.de
Buntware Games	Colmarerstrasse 9	79106 Freiburg	0761 / 3891598	contact@buntware.com
BURNS Games	Danziger Straße 118	10405 Berlin	030 / 40301763	info@burns-games.com
Bright Future	Im Zollhafen 15-17	50678 Köln	0221 / 78982171	info@brightfuture.de
ByteRockers	Schönhauser Allee 36	10435 Berlin	030 / 44318610	info@byterockers.de
Bytro Labs	Stefan-Meier-Straße 8	79104 Freiburg	0761 / 2035462	info@bytro.com
Caipirinha Games	Kiefernweg 1a	24161 Altenholz	0431 / 324540	info@caipirinhagames.de
Candygun Games	Hamburger Strasße 132	22083 Hamburg	040 / 20947878	
cerasus.media	Rosenthaler Straße 51	10178 Berlin	030 / 802089990	info@cerasus.de
Chimera Entertainment	Augsburger Straße 8	80337 München	089 / 28979940	info@chimera-entertainment.com
CipSoft	Gabelsbergerstraße 11	93047 Regensburg	0941 / 6308280	kontakt@cipsoft.com
Claw Design	Ludwigstraße 8	72474 Winterlingen	07434 / 3931	info@clawdesign.com
ClockStone STUDIO	Claudiastraße 14	6020 Innsbruck	+43 / 512 / 890377	info@clockstone.com
Cooee GmbH	Opelstraße 10	67661 Kaiserslautern	06301 / 703290	info@clubcooee.com
Coreplay	Schleissheimer Straße 6-10	80333 München	089 / 57957526	info@coreplay.de
Cranberry Production	Podbielskistraße 166	30177 Hannover	0551 / 76170000	mail@cranberry-production.com
CreaTeam Software	Sünderup 6	24943 Flensburg	0461 / 64808	
Crenetic	Kreuzstraße 3	45468 Mülheim an der Ruhr	0208 / 44421 42	info@crenetic.de
crew51	Saarwerdenstraße 8a	40547 Düsseldorf	0221 / 27228650	info@crew51.com
Crimson Cow	Tarpen 40	22419 Hamburg	040 / 59355237	info@crimsoncow.de
Crytek	Hanauer Landstraße 523	60386 Frankfurt am Main	069 / 21977660	info@crytek.de
Daedalic Entertainment	Papenreye 53 – Workport Unit 2	22453 Hamburg	040 / 43261270	info@daedalic.de
Decane	Hallerstraße 5e	20146 Hamburg	040 / 42910947	info@decane.net
Deck13 Interactive	Baseler Straße 46	60329 Frankfurt am Main	069 / 71671660	info@deck13.com
Destrax Entertainment	Merkurstraße 10	33739 Bielefeld		info@destrax.de
dreiNET	Bergedorfer Straße 96	12623 Berlin		philip@dreinet.org
Dutyfarm	Oranienstraße 6	10997 Berlin	030 / 30368430	info@dutyfarm.com
DynamicHead	Hauptstraße 6	55283 Nierstein Schwabsburg	061 / 33572494	timo.weisse@dynamichead.com
EA Phenomic	Binger Straße 38	55218 Ingelheim	06132 / 783510	info@phenomic.net
Egosoft	Heidestraße 4	52146 Würselen	02405 / 4239970	egoinfo@egosoft.com
Eku Interactive	Saalburgstraße 104	61350 Bad Homburg	06172 / 597727	office@eku.de
Enter-Brain-Ment	Littenstraße 106/107	10179 Berlin		interesse@enter-brain-ment.com
eos interactive	Carl-Zeller Straße 16	75015 Berlin	07252 / 957757	info@eosinteractive.de
exDream	Hildesheimer Straße 5	30169 Hannover	0511 / 37077675	info@exdream.com
Exozet Games	Rotherstraße 20	10245 Berlin	030 / 2465600	games@exozet.com
extra toxic	Hanauer Landstraße 196a	60314 Frankfurt am Main	069 / 48440040	info@extratoxic.com
Fakt Software	Rosa-Luxemburg-Straße 76	8058 Zwickau	0375 / 2000557	info@fakt-software.de
Fancy Bytes	Niederstraße 5	26931 Huntorf	02151 / 3259912	biz@fancy-bytes.de
FDG Entertainment	Agnes-Pockels-Bogen 1	80992 München	089 / 18941370	info@fdgentertainment.com
Freudenreich & Tollknaepper	Paracelsusstraße 57	42283 Wuppertal	0202 / 2721048	mail@german-railroads.com
Funatics	Essener Straße 3	46047 Oberhausen	0208 / 43927060	info@funatics.de
funworld	Photo Play Straße 1	A-4860 Lenzing	+43 / 7662 / 67850	info@photoplay.com
Fusionsphere Systems	Rosenau 8	86551 Aichach	0700 / 38746600	info@fusionsphere.com
GameArt Studio	Bismarckstraße 107	10625 Berlin		kontakt@GameArtStudio.de
GameDevelopment M. Molkenthin	Lerchenstraße 98	22767 Hamburg	040 / 46773207	gamedev@michael-molkenthin.de
GameDuell	Greifswalder Straße 212-213	10405 Berlin	030 / 288768211	impressum@gameduell.de
Gameforge	Albert-Nestler-Straße 8	76131 Karlsruhe	0721 / 3548080	info@gameforge.de
gameus	Kufsteinerstraße 40	83064 Raubling	08035 / 6689	info@gameus.de
Gaming Minds Studios	Georgstraße 6	33330 Gütersloh		contact@gamingmindsstudios.com
Gbanga	Brauerstraße 31	8004 Zürich	+41 / 43 / 5366701	matthias@gbanga.com
GIANTS Software GmbH	Badenerstraße 621	8048 Zürich	+41 / 44 / 5082176	mail_us@giants-software.com
Goodgame Studios	Theodorstraße 42-90, Haus 4a	22761 Hamburg	040/ 85192490	info@goodgamestudios.com
Greentube	Mariahilfer Straße 47/1/102	A-1060 Wien	+43 / 1 / 4945056	office@greentube.com
HandyGames	i_Park Klingholz 13	97232 Giebelstadt	09334 / 97570	info@handy-games.com
Independent Arts Software	Münsterstraße 5 / HAMTEC Haus 2b	59065 Hamm	02381 / 688292	info@independent-arts-software.de
Inflammables	Vangerowstraße 16/1	69115 Heidelberg	06221 / 7254864	info@inflammables.de
InnoGames	Harburger Schloßstraße 28	21079 Hamburg	040 / 78893350	info@innogames.de
Instance Four	Friederikastraße 65	44789 Bochum	0234 / 89038971	info@instancefour.com
Island Games	Vahrenwalder Straße 7	30165 Hannover	0511 / 35319898	kontakt@islandgames.de
K1010 Media	Erkelenzdamm 59-61	10999 Berlin	030 / 57709936	info@k1010.de
Keen Games	Dreieichstraße 59	60594 Frankfurt am Main	069 / 66058838	info@keengames.com
King Art	Martinistraße 47-49	28195 Bremen	0421 / 322760	contact@kingart.de
KORION Simulation	Mömpelgardstraße 16	71640 Ludwigsburg	0700 / 56746600	info@korion.de
kr3m.media	Kaiserstraße 158	76133 Karlsruhe	0721 / 18395918	
Kritzelkratz 3000	Dieselstraße 2-6	97082 Würzburg	0931 / 3536091	info@kritzelkratz.de

DEVELOPMENT

Firma	Straße/Hausnummer	Postleitzahl/Ort		Telefon	E-Mail
kunst-stoff GmbH	Rungestraße 22-24	10179	Berlin	030 / 23457178	info@kunst-stoff.de
Limbic Entertainment	Heinrich-Hertz-Straße 21	63225	Langen	06103 / 5019780	info@limbic-entertainment.de
Madcat Interactive Software	Prager Ring 4-12	66482	Zweibrücken	06332 / 791320	info@madcat.de
MaDe Games GbR Neetix - Wycislik	Hochstraße 34	41334	Nettetal		m.wycislik@madegames.com
marune	Wallbergstraße 16	82008	Unterhaching	089 / 66500749	info@marune.de
Master Creating	Winterhuder Weg 82	22085	Hamburg	040 / 33313780	mail@master-creating.com
Media Seasons	Liviastraße 6	4105	Leipzig	0341 / 2156752	contact@mediaseasons.com
mobivention GmbH	Albin-Köbis-Straße 8	51147	Köln	02203 / 906020	
netmin games	Philipp-Reis-Straße 6	55129	Mainz	06131 / 507896	schreiber@netmin.de
Neutron Games	Gustav-Meyer-Allee 25	13355	Berlin	030 / 46307222	kontakt@neutron-games.de
northworks	Stahltwiete 23	22761	Hamburg	040 / 78107075	info@northworks.de
NovaTrix / Animedo	Schlüsselwiesen 19	70186	Stuttgart	0711 / 3102770	info@animedo.de
Nurogames GmbH	Schaafenstraße 25	50676	Köln	0221 / 3988040	info@nurogames.com
OG-Soft Productions	Zirbelstraße 48	86154	Augsburg		info@og-soft.de
online4ever Limited	Manitiusstraße 6	1067	Dresden	0351/ 2063875	info@online4ever.biz
Outline Development	Rappenweiher 12	57074	Siegen	0271 / 3131331	webmaster@outline-development.de
Piranha Bytes	Ruhrallee 63	45138	Essen	0201 / 806720	info.nospam@piranha-bytes.com
PixelPlaying	Vorm Eickerberg 1a	42929	Wermelskirchen	02196 / 8899693	info@pixelplaying.de
pixeltamer.net	Clausewitzstraße 6	10629	Berlin	030 / 34347690	welcome@pixeltamer.net
Pixon	Jaffestraße 10	21109	Hamburg	040 / 33399430	welcome@pixon.de
Playa Games	Winterhuder Weg 82	22085	Hamburg	040 / 33313822	info@playa-games.com
Playnik	Bielefelder Str. 5	90425	Nürnberg	0700 / 42300000	info@adrastea.com
Procontis	Am Bauhof 18	64807	Dieburg	06071 / 430111	sales@procontis.de
Promotion Software	Karlstraße 3	72072	Tübingen	07071 / 91670	mail@promotion-software.de
qforge	Südwestpark 37 - 41	90449	Nürnberg	0911 / 3092097	info@qforge.de
Quadriga Games	August-Bebel-Straße 27	14482	Potsdam	0331 / 23189660	info@quadrigagames.com
Rabenauge	Ebelingstraße 17	10249	Berlin	030 / 21092345	the_contact@rabenauge.com
Rainfire Entertainment	Frohnhauserstraße 65	45127	Essen	0201 / 94613403	info@rainfire-entertainment.com
Ravensburger Digital	Agnes-Pockels-Bogen 1	80992	München	089 / 12228180	tellme@ravensburger-digital.com
Reakktor Media	Kleine Duewelstraße 21	30171	Hannover	0511 / 540980	info@reakktor.com
Realmforge Studios	Hofer Straße 15	81737	München	089 / 55069134	info@realmforgestudios.com
Related Designs	Römerpassage 1	55116	Mainz	06131 / 554470	info@related-designs.de
Silent Future	Vohwinkeler Straße 51-53	42329	Wuppertal	0202 / 5143480	info@silentfuture.de
Snowstep Development	Heinrich-Holtschneider-Weg 76	40489	Düsseldorf	0201 / 17848176	business@snowstep.com
Solimedia Productions	Schlösserstraße 28	99084	Erfurt	0361 / 658550	hatmut.niemeier@solimedia.de
Spellbound	Weststraße 15	77694	Kehl am Rhein	07851 / 991660	info@spellbound.de
SpinBottle Games	Weidkamp 3	45355	Essen	0201 / 1789614	info@spinbottlegames.com
Spirit	Friedrichstraße 15	70174	Stuttgart	0711 / 96685510	weyrich@go4spirit.com
Splitscreen Studios	Neuer Kamp 30	20357	Hamburg		info@splitscreenstudios.com
spotsonfire	Hongkongstraße 7	20457	Hamburg	040 / 209346330	info@spotsonfire.com
Springwald Software	Alter Eistreff 36	44789	Bochum	0700 / 777464925	info@springwald.de
Sproing	Fernkorngasse 10	A-1100	Wien	+43 / 1 / 60430280	office@sproing.com
Synetic	Auf der Haar 47	33332	Gütersloh	05241 / 4009900	synetic@synetic.de
Ticking Bomb Games	Böckmannstraße 15	20099	Hamburg	040 / 80793499	info@ticking-bomb-games.de
Travian Games	Wilhelm-Wagenfeld-Straße 22	80807	München	089 / 3249150	mail@traviangames.com
upjers	Memmelsdorfer Straße 250	96052	Bamberg	0951 / 3015835	mail@upjers.com
VIS Games and Entertainment	Am Keuschenend 49-51	50170	Kerpen	02273 / 999860	info@vis-games.de
Vulcando Games	Dasinger Straße 2	86165	Augsburg	0821 / 4550834	info@vulcando-games.de
wooga	Saarbrücker Straße 38	10405	Berlin	030 / 32505294	info@wooga.com
Xybris Interactive	Neuendorfstraße 16d	16761	Hennigsdorf	03302 / 2085420	info@xybris.com
YAGER Development	Pfuelstraße 5	10997	Berlin	030 / 69597660	contact@yager.de
Zeroscale	Schwedter Straße 9a	10119	Berlin	030 / 405057929	info@zeroscale.com
Zone 2 Media	Am Wasserturm 6	40668	Meerbusch	02150 / 705660	info@zone2.de
Z-Software	Joseph-von-Fraunhoferstraße 20	44227	Dortmund	0231 / 9700780	contact@z-software.net
Zynga Game Germany	An der Welle 4	60322	Frankfurt		rocco@zynga.com

BUSINESS SERVICE

Firma	Straße/Hausnummer	Postleitzahl/Ort		Telefon	E-Mail
4-REAL INTERMEDIA (Lokalisation)	Ludwigstraße 32	63067	Offenbach	069 / 8090880	info@4-real.com
Anakan (Lokalisation)	Pfuelstraße 5	10997	Berlin	030 / 531420450	ana@anakan.de
Aruba Events	Kreuzstraße 1-3	45468	Mülheim an der Ruhr	0208 / 6982608	infos@aruba-events.de
Atlas Interactive (Payment Solutions)	Christoph Probst Weg 3	20251	Hamburg	040 / 413300185	sales@atlasinteractive.de
Dr. Behrmann & Härtel Rechtsanwaltsgesellschaft	Kantstraße 98	10627	Berlin	030 / 319984910	info@behrmannhaertel.de
Between Us! (PR, Marketing)	Wendelsteinstraße 11	85540	Haar	089 / 43906293	
Buschbaum Media & PR	Emanuel-Leutze Straße 21	40547	Düsseldorf	0211 / 5988140	presse@buschbaum-media.com
Compad Company (Merchandising)	Habsburger Straße 85	79104	Freiburg	0761 / 5853667	info@com-pad.com
deal united (Monetization solutions)	Landshuter Allee 10	80637	München	089 / 381649360	info@dealunited.com
Friedmann Kommunikation (Marketing / PR)	Schanzenstraße 36	51063	Köln	0221 / 5348710	info@fr-k.de
gamescom / koelnmesse (PR, Marketing)	Messeplatz 1	50679	Köln	0221 / 8213894	info@koelnmesse.de
GAN Game Ad Net	Stresemannstraße 342	22761	Hamburg	040 / 67586750	
Ganz & Stock (Personnel Consulting)	Westhafentower, Westhafenplatz	60327	Frankfurt am Main	069 / 710456380	a.stock@ganz-stock.de
Gärtner PR (PR, Marketing)	Destouchesstraße 68	80796	München	089 / 30766854	info@gaertner-pr.de
Game Developers Conference Europe	Kaiser-Wilhelm-Straße 30	12247	Berlin	030 / 34622644	nicolas.fluschnik@ubm.com
GRAEF Rechtsanwälte	Jungfrauenthal 8	20149	Hamburg	040 / 80600090	www.graef.eu

BUSINESS SERVICE

Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
IDG Entertainment Media GmbH (Custom Publishing)	Lyonel-Feininger-Straße 26	80807 München	089 / 360860	dsteiger@idg.de
IGA Worldwide Ltd. (In-Game-Advertising)	Oranienburger Straße 86a	10178 Berlin	030 / 2008900	info@ingameadvertising.com
IHDE & Partner (Rechtsberatung)	Schoenhauser Allee 10-11	10119 Berlin	030 / 44318660	berlin@onlinelaw.de
IMC (Full-Service Dienstleister)	Hauptstraße 172	51143 Köln	02203 / 949092	info@imc-online.net
Indigo Pearl (PR, Marketing)	Richardstraße 45	22081 Hamburg	040 / 25330279	pr@indigopearl.de
INPROMO (PR-Agentur)	Banksstrasse 2b	20097 Hamburg	040 / 23881771	kontakt@inpromo.de
Keyfactor (Consulting)	Rudolf-Diesel-Straße 20a	65760 Eschborn	06173 / 999781	contact@keyfactor-entertainment.de
Marchsreiter Communications (PR, Marketing)	Guldeinstraße 41a	80339 München	089 / 51919942	dm@marchsreiter.com
ME Enterprises (Consulting)	Herrenhäuser Straße 24	30938 Burgwedel	05139 / 9599915	info@meenterprises.de
Osborne Clarke (Anwaltskanzlei)	Innere Kanalstraße 15	50823 Köln	0221 / 51084000	info@osborneclarke.de
Partnertrans (Lokalisation)	Leipziger Straße 28	40822 Mettmann	02104 / 172660	info@partnertrans.com
Play + Smile Marketing (PR, Marketing)	Am Wasserturm 6	40668 Meerbusch	02150 / 705660	info@play-and-smile.de
ProCom International LTD	Kisdorfer Weg 3b	24568 Kaltenkirchen	040 / 71669763	info@procom-international.com
Quinke Networks (PR, Marketing)	Bei den Mühren 70	20457 Hamburg	040 / 43093949	mail@quinke.com
remote control productions (Producing, Consulting)	Augsburgerstraße 8	80337 München	089 / 26949648	info@r-control.de
Rode + Mathé (Rechtsberatung)	Henriettenweg 4	20259 Hamburg	040 / 43270432	info@rodemathe.de
ROESSLER PR (Marketing, PR, Consulting)	Walter-Leiske-Straße 2	60320 Frankfurt am Main	069 / 514461	communicate@roesslerpr.de
SCC (Crossmedia)	Orleansstraße 5a	81669 München	089 / 33094660	info@scc-feld.de
Schanz International (Consulting)	Bgm.-Oberhettinger-Straße 2a	67146 Deidesheim	06326 / 6010	info@schanzgames.com
Schulte Riesenkampff (Rechtsberatung)	Hochstraße 49	60313 Frankfurt am Main	069 / 900266	schulte@schulte-lawyers.de
Selected Minds (Personalberatung)	Walther-Rathenau-Straße 16	64521 Groß-Gerau	06152 / 1871830	info@selected-minds.de
Sputnic Consulting	Ehrenbergerstr. 11	98693 Ilmenau	03677 7996990	media@sputnic-consulting.com
Station54 Community Management	Pankstraße 8-10	13127 Berlin	030 / 44716415	info@station54.de
swordfish pr (PR, Marketing)	Kaiserstraße 7	80801 München	089 / 60031641	ilka@swordfish-pr.de
Tom Putzki Consulting	Behrensstrasse 43	44623 Herne	02323 / 9190053	tp@tom-putzki-consulting.com
Translation-Taxi	Beusselstraße 28	10553 Berlin	030 / 28371428	info@translation-taxi.com
Turtle Entertainment (eSport)	Siegburger Straße 189	50679 Köln	0221 / 8804490	info@turtle-entertainment.de
Two Pi Team (Community Management, Lokalisation)	Gerresheimer Straße 9	40721 Hilden	02103 / 9411914	info@two-pi-team.de
Walk Game Productions (Produktion)	Karl-Leopold-Straße 6	76229 Karlsruhe	0721 / 48399963	info@wolfgangwalk.de

TECHNICAL SERVICE

Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Akamai Technologies (Datentransfer-Technologie)	Betastraße 10B	85774 Unterföhring	089 / 94006308	contact-germany@akamai.com
Artificial Technology (KI-Middleware)	Dornierstraße 2	82178 Puchheim	089 / 80076124	info@artificial-technology.com
arvato games task force (Full-Service-Dienstleister)	Carl-Bertelsmann-Straße 161f	33311 Gütersloh	05241 / 803074	kontakt@gamestaskforce.de
BiteTheBytes (Middleware)	St. Gallener Straße 9	36039 Fulda	0661 / 2006899	info@bitethebytes.com
B.TON Medien (Audio, Musik)	Luise-Ullrich-Straße 4	82031 Grünwald	089 / 189425490	mix@bton.de
Centroid (Motion Capturing)	Schlesische Straße 27	10997 Berlin	030 / 698074 74	sven.bergmann@centroidanimation.com
Chromatrix (Mobile Gaming)	Kalkofenstraße 2	72411 Bodelshausen	07471 / 740828	info@chromatrix.com
Codecult (3D-Engine)	Kemnastraße 21a	44866 Bochum	02327 / 35474	info@codecult.com
ContenTainer (Leveldesign)	Eichelkopfstraße 17	63584 Gründau	06058 / 9178150	info@contentainer.de
Creature Factory	Graefestraße 33	10967 Berlin	030 / 26301330	info@creature-factory.com
Crytek (3D-Engine)	Hanauer Landstraße 523	60386 Frankfurt am Main	069 / 21977660	cryengine@crytek.com
definitv:ton (Audio)	Mozartstraße 35-37	50674 Köln	0221 / 8234220	mhoechsmann@definitiv-ton.de
Die Kolonie (Art, Design)	Adelheidstraße 82	65185 Wiesbaden	0611 / 5058765	contact@diekolonie.com
Dynamedion (Sounddesign)	Barbarossaring 8	55118 Mainz	06131 / 5847895	info@dynamedion.com
DynamicHead (Unity Services)	Hauptstraße 6	55283 Nierstein Schwabsburg	061 / 33572494	timo.weisse@dynamichead.com
Effective Media (Lokalisation)	Schwarzmannstraße 8	80798 München	089 / 38998999	info@effective-media.de
Eye Rock Media (Design, Animation)	Essenerstraße 3	46047 Oberhausen	0208 / 82858923	contact@eyerock.de
fadeout (Audio Design)	Austraße 6	CH-4153 Reinach	+41 / 7 / 92793244	info@fadeout.ch
fatfoogoo (Online-Payment, Store-Solutions)	Mariahilferstraße 50	A-1070 Wien	+43 / 1 / 23622970	office@fatfoogoo.com
Flow Studios	Gustav-Meyer-Allee 25	13355 Berlin	030 / 8092920	info@flow-studios.eu
Games Quality (QA)	Neu Zittauer Straße 15	15537 Erkner	03362 / 881207	info@games-quality.com
Global Producer (Production)	Hauptstraße 172	51143 Köln	02203 / 97722400	getintouch@global-producer.com
Glare Studios (Grafik)	Mansteinstraße 18	20253 Hamburg	02307 / 4388404	contact@glarestudios.de
GreenMamba-Studios (Animation)	Aloysstraße 7	48249 Dülmen	02594 / 7928217	info@greenmamba-studios.de
Hans HiScore (Audio)	Friedrich-Ebert-Anlage 11-13	60327 Frankfurt	0151 / 11616133	info@hans-hiscore.de
Intulo (3D, Animation)	Lindholz 89	31139 Hildesheim	05121 / 2944022	info@intulo.de
keuthen.net IT & Network Solutions	Neue Mainzer Straße 76-78	55129 Mainz	06131 / 9459880	info@keuthen.net
Lingoona (Loklisations-Middleware)	Hildastraße 38	68723 Plankstadt	06202 / 4095030	info@lingoona.com
metricminds (Grafik)	Rüsselsheimer Straße 22	60326 Frankfurt am Main	069 / 7593380	info@metricminds.com
modus werbung (PR, Marketing)	Neuer Kamp 30, Eingang C	20357 Hamburg	040 / 2380560	mail@modus-werbung.de
morro images (Grafik)	August-Bebel-Straße 27	14482 Potsdam-Babelsberg	0331 / 97996610	info@morroimages.com
Nevigo (Tools & Middleware)	Lyrenstraße 13	44866 Bochum	02327 / 8369840	info@nevigo.com
Periscope Studio	Am Diebsteich 55	22761 Hamburg	040 / 31811767	info@periscopestudio.de
Phenomatics (XNA-Entwicklung, Consulting)	Hafenstrasse 47-51	A-4020 Linz/Donau	+43 / 732 / 90155230	office@phenomatics.com
Polyce (Grafik)	Littenstraße 106/107	10179 Berlin	0178 / 3592560	info@polyce.de

TECHNICAL SERVICE

Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Port Plexus (Special Edition Design)	Am Varrenbruch 8	29690 Schwarmstedt	05071 / 96850	info@portplexus.com
Rabcat Computer Graphics (Grafik)	Fernkorngasse 10	A-1100 Wien	+43 / 1 / 5237425	mail@rabcat.com
Razer (Hardware)	Winterhuder Weg 82	22085 Hamburg	040 / 419299300	
Rocketbox Studios (Grafik)	Leonhardtstraße 10	30175 Hannover	0511 / 8984384	info@rocketbox.de
Spinor	Agnes-Pockels-Bogen 1	80992 München	089 / 54043980	spinor@spinor.com
The Light Works (Grafik)	Otto-Hahn-Straße 7	50997 Köln	02236 / 967322	info@thelightworks.com
ToBringAlive (Animation, Design)	Ostmarkstraße 49	48145 Münster	0251 / 392312	angerbauer@tobringalive.com
Tektroop (Support)	Vogelsangstrasse 36	CH-8133 Esslingen	+41/ 43 / 2776040	info@tektroop.ch
TON & SPOT audiodesign (Sound)	Immanuelkirchstraße 15	10405 Berlin	030 / 92129727	post@ton-und-spot.de
Toygardens Media (Grafik, Video)	Walderseestraße 54	30163 Hannover	0511 / 33659800	akunze@toygardens.com
Trinigy (3D-Engine)	Arbachtalstraße 6	72800 Eningen u.A.	07121 / 986993	info@trinigy.de
VCC Perfect Pictures (Grafik, Video)	Doormannsweg 43	20259 Hamburg	040 / 431690	hamburg@vcc.de
Versant (Data Management)	Wiesenkamp 22b	22359 Hamburg	040 / 609900	info@versant.com
Virgin Lands (Grafik, Cinematics)	Hauptstraße 5	97332 Volkach	09381 / 71780	info@virgin-lands.com
xaitment	Fischbacherstraße 92	66287 Quierschied	06897 / 600800	info@xaitment.com
Z-Ground Illustration (Grafik)	Oeltzenstraße 17	30169 Hannover	0511 / 1696959	jz@z-ground.com
zuuka! (Audio, Lokalisation)	Christian-Pless-Straße 11-13	63069 Offenbach am Main	069 / 2475700	mail@zuuka.de

EDUCATION

Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Akademie für Medien	Balinger Straße 39A	70567 Stuttgart	0711 / 4209444	h.ehlers@amk-net.de
Cologne Game Lab / Fachhochschule Köln	Ubierring 40	50678 Köln	0221 / 82753095	info@colognegamelab.de
Design Schule Schwerin	Bergstraße 40	19055 Schwerin	0385 / 5559775	info@designschule-schwerin.de
FH Trier	Schneidershof	54293 Trier	0651 / 8103345	sek@informatik.fh-trier.de
FH Salzburg	Urstein Süd 1	A-5412 Puch/Hallein	+43 / 50 / 22110	press@fh-salzburg.ac.at
Games Academy Berlin	Rungestrasse 20	10179 Berlin	030 / 29779120	info@games-academy.de
Games Academy Frankfurt	Hanauer Landstraße 146	60314 Frankfurt	069 / 42696460	info@games-academy.de
HTW Berlin	Wilhelminenhofstraße 75a	12459 Berlin	030 / 50190	
Macromedia Hochschule	Gollierstraße 4	80339 München	089 / 5441510	info.muc@macromedia.de
Mediadesign Hochschule	Lindenstraße 20-25	10969 Berlin	030 / 3992660	info-ber@mediadesign-fh.de
Mediadesign Hochschule	Werdener Straße 4	40227 Düsseldorf	0211 / 1793930	info-dus@mediadesign-fh.de
Mediadesign Hochschule	Claudius-Keller-Straße 7	81669 München	089 / 4506050	info-muc@mediadesign-fh.de
Qantm Institute	Unterhachinger Straße 75	81737 München	089 / 89068771	muenchen@qantm.de
Qantm Institute	Soltauer Straße 18-22	13509 Berlin	030 / 43094470	berlin@qantm.de
Qantm Institute	Linke Wienzeile 130A	A-1060 Wien	+43 / 1 / 961 03 03	andreas@sae.at
SRH Fachschulen	Bonhoefferstraße 1	69123 Heidelberg	06221 / 884225	it.heidelberg@fachschulen.srh.de
SRH Hochschule Heidelberg	Ludwig-Guttmann-Straße 6	69123 Heidelberg	06221 / 882790	tobias.breiner@fh-heidelberg.de
Zürcher Hochschule der Künste (ZHdK)	Ausstellungsstraße 60	CH-8005 Zürich	+41 43 / 4464646	info.ddk@zhdk.ch

VERBÄNDE

Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
BIU	Rungestraße 18	10179 Berlin	030 / 24087790	info@biu-online.de
Cluster Audiovisuelle Medien Bayern	Sonnenstraße 21	80331 München	089 / 54460240	info@cam-bayern.de
G.A.M.E.	Schönhauser Allee 10-11	10119 Berlin	030 / 44318668	info@game-bundesverband.de
gamearea FRM	Holzhausenstraße 22	60322 Frankfurt am Main	069 / 94419441	info@gamearea-frm.de
gamecity:Hamburg	Habichtstraße 41	22305 Hamburg	040 / 2270190	info@gamecity-hamburg.de
GAMEplaces	Hanauer Landstraße 126-128	60314 Frankfurt am Main	069 / 21236214	manuela.schiffner@gameplaces.de
Hessen-IT	Abraham-Lincoln-Straße 38 - 42	65189 Wiesbaden	0611 / 7748481	info@hessen-it.de
Mitteldeutsche Medienförderung	Hainstraße 17-19	4109 Leipzig	0341 / 269870	info@mdm-online.de
nordmedia	Expo-Plaza 1	30539 Hannover	0511 / 1234560	info@nordmedia.de
Northstar Developers	Bödekerstr 84	30161 Hannover	0160 / 97880272	d.bruenig@collision-studios.com
ÖVUS	Aichhorngaße 14	A-1120 Wien	+43 / 1810 / 328912	office@ovus.at
USK	Marchlewskistraße 27	10243 Berlin	030 / 293638290	kontakt@usk.de

FREELANCER

Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Andreas Adler (Audio)	Wernigeroder Weg 18	30419 Hannover	0511 / 3889163	andreas@adler-audiopictures.de
Anja Heppelmann (Übersetzung)	Kramers Kamp 1a	48291 Telgte	02504 / 7399601	anja@spielworte.de
Anne-Petra Lellwitz (Marktforschung)	Fichtestrasse 65	63303 Dreieich	0160 / 2014364	info@apl-mafo.de
Arwed Ralf-Grenzbach (Consulting)	Zionskirch Straße 8	10119 Berlin	0171 / 6486720	arg@conflutainment.com
Daniel Pharos (Audio)	Golddistelanger 15	80937 München	0179 / 5210100	daniel@knightsofsoundtrack.com
Falko Löffler (Autor)	Lindenstraße 8a	36355 Grebenhain-Ilbeshausen	06643 / 918577	mail@falkoloeffler.com
Grafikbüro Werner (Grafik)	Beethovenstraße 40a	48317 Drensteinfurt	02508 / 985801	info@grafikbuero-werner.de
Janine Smitkiewicz (Rechtsanwältin)	Karlsplatz 7	80335 München	089 / 45244350	smitkiewicz@auer-smitkiewicz.de
Jochen Hamma (Producing)	Wehinger Weg 9	78583 Boettingen	07429 / 910793	jochen.hammer@fantastic-realms.com
Jörg Beilschmidt (Game Design)	Diemelkoppel 5	21406 Melbeck	04134 / 900860	j.beilschmidt@creatown.de
Marc Huppke (Projektmanagement)	Frankfurter Straße 5	64521 Gross-Gerau	06152 / 9614432	marc@make-projects.com
Marc Oberhäuser (Game Design, Consulting)	Berzbuirer Strasse 98	52355 Düren	02421 / 51547	marc.oberhauser@freelancer-games.com
Martin Deppe (Producing, Game Design)	Mozartstraße 1	97232 Giebelstadt	09336 / 979653	martin.deppe@wortspielmacher.de
Maurice T. Anderson (Sounddesign)	Klosterstraße 3	63486 Bruchköbel	0176 / 26228136	contact@maurice-t-anderson.com
Michael Bhatty (Entwicklung, Consulting)	Bismarckstraße 24	49076 Osnabrück	0541 / 1810660	michael@bhatty.de
Nico Nowarra (Storytelling)	Friedbergstraße 9	14057 Berlin	030 / 31801899	nno@menintext.com
Nicole Müller (PR, Marketing, Events)	Glyzinenstraße 7c	80935 München	0173 / 7872013	nm@gconsultnet.de
Niels Bauer Games (Game Design)	Sickingenstraße 37	79117 Freiburg	0761 / 66947	contact@nielsbauergames.com
Sabrina Heuer (Audio Production)	Bachemer Straße 191	50935 Köln	0163 / 8604004	sabrina@froschtatze.de
Stefan Köhler (Lektor / Autor)	Danziger Straße 9	38642 Goslar	0176 / 22512273	stefanclemenskoehler@web.de

Wenn ihr wüsstet ...
... warum Lokalisierung wirklich schiefgeht!

Neue Serie: Die Games-Branche ist super, keine Frage. Dennoch gibt es viele Dinge, über die man sich wunderbar und völlig zu Recht aufregen kann. Zum Auftakt berichtet ein Übersetzer über die Schattenseiten seines Jobs.

Bei allem verständlichen Ärger über schlechte Lokalisierungen von Videospielen: Ganz oft ist das gar nicht anders zu machen! Ganz oft sorgt der Publisher bzw. Hersteller durch seine Strukturen, Prozesse und realitätsfernen Vorstellungen dafür, dass die Sprachversionen nichts anderes werden können als miserabel. Das tut er natürlich nicht absichtlich, doch die Lokalisierung scheint immer das fünfte Rad am Wagen zu sein. Natürlich notwendig, aber irgendwie auch lästig, grundsätzlich immer zu teuer und vor allem: zu zeitaufwendig.

Mal ein Beispiel, das vielleicht übertrieben klingt, aber gar nicht ist: »Game X« ist komplex, hat circa 300.000 Wörter Text, wird nicht vertont, sondern untertitelt. Dauer der Lokalisierung: vier Monate inkl. Loka-QA und Testing.

»Game X« wird ein großer Erfolg, vier Jahre später kommt »Game X – Part YZ« heraus. »Game X – Part YZ« ist weit komplexer als der Vorgänger, hat nun knapp 900.000 Wörter Text im Hauptspiel, wird ebenfalls nur untertitelt, hat aber – ganz neu – eine Community-Schnittstelle mit täglich wechselnden Inhalten und bergeweise herunterladbare Inhalte mit noch mal circa 600.000 Wörtern, die möglichst mit dem Release des Spiels fertig sein sollen. Dazu kommen Entwicklerblogs, Homepage usw., die – und da besteht der Publisher/Hersteller drauf! – von denselben Leuten übersetzt werden sollen, damit »alles aus einem Guss ist«.

Zeitansatz für diesen Wahnsinn: zwei Monate inklusive Loka-QA und Testing. Darin enthalten ist übrigens schon die scheinbar unvermeidliche Periode des »für die Tonne Schreibens«, denn entgegen anderslautenden Beteuerungen kommen die ersten gelieferten Texte niemals in der letztgültigen Fassung bei der Loka-Agentur an. Spätestens eine Woche nach Projektstart trudeln überarbeitete Fassungen ein, und das bis dahin Geschriebene ist für die Katz. Nach Projektende darf man sich dann dafür rechtfertigen, warum man die Dreistigkeit besitzt, diese Arbeit auch noch in Rechnung zu stellen.

Wie geht eine Loka-Agentur solchen Wahnsinn am besten an? Ganz einfach: Sie verhaftet augenblicklich jeden freien und billigen Übersetzer und verteilt Textportionen, die nicht sortiert oder sonst wie gegliedert sind, zum Beispiel nach Missionen. Wenn es sich blöd überlappt, schreiben drei Übersetzer an derselben Mission rum, ohne voneinander zu wissen. Und vor allem: ohne das Spiel zu kennen! Denn die Zeiten, in denen man seinen eigenen Gamebuild bekam, um übersetzte Passagen selbst im Spiel testen zu können, sind längst vorbei. Meist gibt es nur zweimal pro Woche die Möglichkeit, ein Excel-Dokument mit Fragen an die Entwickler zu schicken. Hin und wieder werden diese Fragen auch zufriedenstellend beantwortet. Manchmal sogar vor Projektende.

Screenshots von Gegenständen mit kryptischen Bezeichnungen? Gott bewahre! Zeit, ein Glossar zu erstellen und in der Folge auch zu pflegen? Null. Zeit, die übersetzten Texte noch einmal sorgfältig zu redigieren? Null.

Und wo wir schon mal beim Redigieren sind: Den Redakteur müsste die Loka-Agentur auf eigene Rechnung beauftragen, der ist gar nicht im Budget vorgesehen. Das hat nämlich irgendwer anhand der Anzahl der zu übersetzenden Wörter festgelegt, und ein Redakteur übersetzt ja nicht. Einen solchen zu beschäftigen, sieht der Publisher/Hersteller meist auch gar nicht ein, denn der Publisher/Hersteller hat irgendwo eine Loka-QA-Abteilung sitzen, die den Texten »den letzten Schliff« verpasst. Heißt in der Praxis: Da sitzen dann Leute, die eine leidlich gute Übersetzung abschließend endgültig zum Teufel schreiben. Wie oft muss man panisch vom Sofa aufspringen und in alten Übersetzungen wühlen, um sich zu vergewissern, dass dieser Schwachsinn nicht aus dem eigenen Hause stammt, sondern irgendwann später reingerutscht sein muss! Natürlich passieren Fehler bei der Übersetzung. Teilweise sogar ganz wilde. Aber wirklich hanebüchene Dinge passieren, wenn jemand etwas ändert, nur um etwas geändert zu haben. Die Jungs und Mädels brauchen schließlich einen Arbeitsnachweis. So eine QA-Abteilung kostet einen Haufen Geld, also wird geschrieben, verschlimmbessert, gekürzt, verdreht und geschwurbelt.

So sehr man sich auch vornimmt, gute Qualität abzuliefern: Oft ist es schlicht nicht möglich!

Mister X

Mister X

ist der Redaktion bekannt.

Paradebeispiel für misslungene Loka: In der deutschen Version des Rollenspiels **Elder Scrolls: Oblivion** wurde der Heilzauber als Feuerball bezeichnet.

Was zu meckern?

Mal ehrlich: Jeder von uns würde doch gern mal so richtig schön vom Leder ziehen, was uns am Job aufregt. Dann tun Sie's doch einfach! Legen Sie den Finger in die Wunde, machen Sie auf Probleme aufmerksam und wecken Sie vielleicht ein wenig mehr Verständnis für Ihren Arbeitsbereich – selbstverständlich anonym.
Schicken Sie uns einfach Ihren Text an ...
info@makinggames.de
Betreff: Wenn ihr wüsstet

Mister X

Was macht eigentlich ...
Richard Bartle

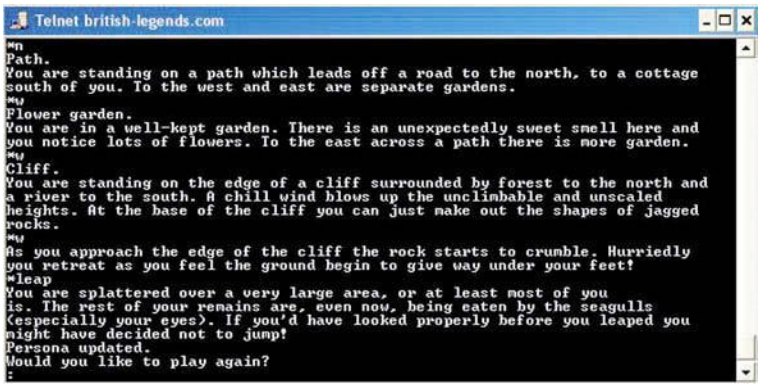
Mit den MUDs hat Dr. Richard Bartle den Grundstein für die ersten Online-Rollenspiele gelegt, seine Erkenntnisse im Bereich der »Spielertypen« gehören zum Einmaleins der Computerspielforschung. Wir haben mit ihm über seine Arbeit und die Evolution der MMOGs gesprochen.

Dr. Richard Bartle

ist Gastprofessor an der Universität Essex und Berater für die Spieleindustrie.

Making Games Was genau machen Sie derzeit?
Richard Bartle Mein berufliches Leben ist zweigeteilt. In der Hälfte der Zeit bin ich Akademiker, in der anderen Hälfte Berater für die Spieleindustrie. Jedes Mal, wenn ich als Akademiker arbeite, denken die Leute, ich sei eigentlich jemand aus der Industrie, der sich eben auch mal als Akademiker versucht. Umgekehrt denken viele bei meiner Beratertätigkeit, ich sei eigentlich ein Akademiker, der auch in Consulting macht. Beide Seiten haben also das Gefühl, ich sei ein Außenseiter, der in ihren Bereich eindringt.
Ich möchte wirklich ein Berater in Vollzeit sein, allerdings ist dies schwierig, wenn die Arbeit nur in unvorhersehbaren Mengen kommt. Mal gibt es drei Monate lang nichts zu tun, dann wieder zwei Wochen lang Vollzeitarbeit, im Anschluss wieder nichts und dann gleich mal ein paar Monate am Stück – das ist dann natürlich schwierig zu planen. Allerdings mache ich deutlich mehr Geld mit der Beratung. Ein paar Tage und ich habe schon die Akademiker-Einnahmen aus einem ganzen Monat drin (lacht).

Making Games Nehmen Sie während der Beratung spezifische Elemente des Spiels unter die Lupe oder schauen Sie sich das Spiel als Ganzes an und geben Ihre Einschätzung?
Richard Bartle Wenn ich bei einem Projekt hinzugezogen werde, gibt es in der Regel ganz bestimmte Felder, denen ich mich widmen soll. Manchmal ist es ganz zu Beginn in der wilden Konzeptphase, wo es um die Evaluierung der Spielidee geht. Mittlerweile erhalte ich oft Designdokumente und bin rund zwei bis drei Wochen damit beschäftigt, diese wirklich durchzuarbeiten. Manchmal stoße ich nach diesen frühen Phasen hinzu und bewerte all das, was das Team schon zusammengebaut hat. Und manchmal werde ich erst in einem späten Stadium gerufen – also genau dann, wenn alles schon schrecklich falsch gelaufen ist. Das ist wirklich schrecklich, weil alle im Unternehmen wissen, was passiert, wenn das Projekt eingestampft wird. Nur traut sich keiner, es auszusprechen. Und dann muss ich jemanden beiseitenehmen und ihm sagen: »Hör zu, du hast wirklich gute Leute hier, die eine Menge leisten, aber ihr braucht etwa einen Vormittag, um ein neues Rüstungsteil zu bauen. Das Problem ist, ihr braucht nicht eines, sondern 100 Rüstungsteile, also benötigt ihr insgesamt 50 Tage, also drei Monate Arbeit, nur um genügend Rüstungen zu erstellen. Und dann braucht ihr noch Pferde, Schiffe usw. Ihr habt einfach nicht die Manpower, um das zu stemmen und werdet es so nicht schaffen. Ihr braucht also entweder einen Haufen Geld oder ihr werdet über kurz oder lang untergehen. Da könnt ihr auch gleich aufhören.« Das ist dann der Zeitpunkt, wenn ich den Leuten sagen muss, was sie eigentlich schon wissen. Aber sie brauchen eben jemanden von außen, der ihnen ihr Gefühl noch einmal bestätigt, was natürlich immer eine traurige Angelegenheit ist. In sehr seltenen Fällen kann ich ihnen aber auch einen Ausweg aufzeigen – einen letzten Anker, der sie noch retten könnte.


Telnet british-legends.com
mn
Path.
You are standing on a path which leads off a road to the north, to a cottage south of you. To the west and east are separate gardens.
mu
Flower garden.
You are in a well-kept garden. There is an unexpectedly sweet smell here and you notice lots of flowers. To the east across a path there is more garden.
mu
Cliff.
You are standing on the edge of a cliff surrounded by forest to the north and a river to the south. A chill wind blows up the unclimbable and unscaled heights. At the base of the cliff you can just make out the shapes of jagged rocks.
mu
As you approach the edge of the cliff the rock starts to crumble. Hurriedly you retreat as you feel the ground begin to give way under your feet!
mleap
You are splattered over a very large area, or at least most of you is. The rest of your remains are, even now, being eaten by the seagulls (especially your eyes). If you'd have looked properly before you leaped you might have decided not to jump!
Persona updated.
Would you like to play again?
:

Das Original-**MUD** (Multi-User-Dungeon) aus dem Jahr 1978. Das textbasierte MMOG gilt als erste virtuelle Welt und damit als Urvater heutiger Online-Rollenspiele.

Making Games Können Sie ein paar Unternehmen nennen, die Sie beraten haben? Sind es nur MMOG-Projekte, die Sie betreuen?
Richard Bartle Früher waren es MMOGs, jetzt sind es vermehrt Facebook- und Browsergames. Ich kann leider keine Firmen beim Namen nennen, aber so viel darf ich sagen: Ich habe nicht an World of Warcraft gearbeitet und an keinem Spiel von Zynga mitgewirkt. Aber ich habe an Produkten gearbeitet, die eine Stufe tiefer liegen. Wenn man sich also die Titel hinter WoW anschaut, dann kann man es schon erahnen.

Making Games Würden Sie sagen, dass es fundamentale Designelemente gibt, die vielleicht schon damals für die MUDs galten und auch heute in modernen MMOGs gleichermaßen

notwendig sind – oder auf der anderen Seite gar nicht mehr funktionieren?
Richard Bartle Es gibt einige Dinge, die gar nicht funktionieren, weil sie nicht skalieren. Wenn man eine Welt mit der Spieleranzahl heutiger MMOGs hat, dann kann man einige Dinge aus den MUDs nicht mehr anwenden. Im Support kann kein einzelner Mitarbeiter mehr zehn Kunden betreuen, man rechne sich das mal auf World of Warcraft hoch. Ein weiteres Problem ist die Grafik bzw. die Räumlichkeit. Wir konnten damals riesige Gebiete bauen, gleichzeitig aber die Reisezeit extrem kurz halten. Bei einer festen räumlichen Ausdehnung wie in heutigen Spielen kann man aber nicht mehr einfach in einer Minute von der einen Seite einer Wüste auf die andere Seite gelangen – das geht nur, wenn die Wüste tatsächlich klein ist. Und wenn man sich das mal in den aktuellen Spielen anschaut, dann merkt man, wie klein im Grunde alles ist. Sturmwind mag zwar groß aussehen, ist aber im Endeffekt gut zu Fuß zu durchqueren. Die Kontinente in World of Warcraft sind eigentlich keine, sie sehen nur so aus, während man damals tatsächlich einen Kontinent bauen konnte.

Making Games Hat sich also auch das »Spielen« an sich in den letzten Jahren verändert?
Richard Bartle Es hat durchaus eine Entwicklung des Spielens gegeben. Um 1989/1990 herum waren die meisten Spiele um das Konzept »Erforschen und Entdecken« aufgebaut. Man hat den Spielern mehr Kontrolle gegeben, es waren vielmehr Sandboxes. Die Social Gamer, die vor allem wegen ihrer Mitspieler in der Welt unterwegs waren, haben sich mit der Zeit abgespalten und sich mehr mit Konzepten wie Second Life auseinandergesetzt. Die übrigen »Hardcore«-Spieler waren damit sozusagen befreit von allen, die ständig sagten: »Du kannst dies nicht machen, du musst das so machen!« Daraus entstanden ist dann DikuMUD, dessen Gameplay ziemlich genau dem entspricht, was man in heutigen Online-Rollenspielen findet. Sie haben allerdings einige Dinge nicht übernommen – manche, weil sie schwierig zu implementieren waren und andere, weil man ihre Bedeutung nicht erkannte. Die alten Text-Spiele sind also viel detaillierter als moderne Spiele. Dazu eine Anekdote aus MUD 2: Dort gibt es einen Taktstock und einen Bogen. Winkt der Spieler mit dem einen, wird er zum anderen teleportiert. Jetzt kam ein Spieler auf die Idee, den Bogen in einen Brunnen zu werfen. Dieser trieb daraufhin durch die Kanalisation und blieb in einem Gitter hängen, das wiederum in einen geheimen Raum führte. Der Spieler winkte mit dem Stock und wurde zum Bogen teleportiert. Er merkte, dass er dort absolut sicher war, weil es nur einen Weg hinein gab. Ein anderer Spieler wiederum schickte ein kleines Boot gefüllt mit Schießpulver und einer Fackel hinterher. Das Boot explodierte unten am Gitter und tötete den anderen Spieler.

Solche kreativen Situationen erleben wir selten in einem modernen MMOG wie World of Warcraft.

Making Games Tobt also ein Krieg zwischen dem Game Designer als Businessmensch und dem Designer als Künstler?
Richard Bartle Ich denke schon. Am Ende ist dein Unternehmen dein Produkt. Du verkaufst es an ein größeres Unternehmen und damit ist es nun mal ein Produkt. Du entwickelst mit einem Spiel also nur etwas, das deinem eigentlichen Produkt Wert verleiht. Deshalb tendieren Unternehmen dazu, sich in Entwicklungs- und Produktionsseite aufzuteilen. Einerseits wird also das Spiel, andererseits die Firma weiterentwickelt. Gerade im Bereich Browsergames merkt man zurzeit den großen Dreh weg von Spielen als Kunst. Es geht nicht um den hübschen Sonnenschirm, den kleinen Tiger oder die Kuh auf der Weide, es geht ums Geld.

Diese Einstellung wird aber auf lange Sicht nicht funktionieren, denn die Leute wollen irgendwann Neues sehen. Es wird also eine Form von Kreativität nötig sein, um die Menschen zu fesseln, und zwar über das hinaus, was man aus A/B-Tests und Ähnlichem erfährt. Besonders in den USA merkt man einen starken Trend zur Industrialisierung von Browsergames. Am Ende wird es aber eine Kunstform bleiben und man braucht einfach die Leute, die die Spiele voranbringen – einfach schon deshalb, weil der Spieler dies wertschätzen wird.

Making Games Ihre Prognose: Welche Zukunft sehen Sie für MMOGs?
Richard Bartle Es gibt drei Wege, die die Branche nehmen könnte: entweder den guten, den schlechten oder den schmutzigen. Bei der schmutzigen Variante verwässern die Spiele, sie werden anspruchslos, einfach um Massen an Leuten ans Spielen zu bringen. Es ist alles seicht und öde und kein Hardcore Gamer wird dabei angesprochen. Es gibt schon jetzt solche MMOGs und ich hoffe einfach, dass die Spieler sich lieber andere Dinge anschauen.

Der schlechte Weg wäre, wenn die Regierungen auf den Geschmack kommen würden und an all das Geld wollten, das verdient und nicht besteuert wird. Man denke nur an die ganzen Ingame-Währungen, die man vielleicht schröpfen könnte. Oder sie sagen sich »Hey, schaut euch all die Kinder an, die Shooter spielen und dann in der Schule rumballern!« oder »Schaut euch die ganzen Leute an, die spielen und lieber fernsehen sollten!«

Der dritte und gute Weg wäre schließlich, wenn die Entwicklungskosten sinken und damit viel mehr Menschen in die Lage versetzt würden, selbst Spiele zu entwickeln. Es wäre doch fantastisch, wenn drei bis vier Leute ein tolles Produkt auf den Markt bringen könnten, an das vorher noch niemand gedacht hat – oder wenn die eigenen Kinder selbst virtuelle Welten erstellen und in ihnen spielen könnten.

Making Games Dann hoffen wir mal auf die Variante 3. Vielen Dank für das Gespräch!

Interview: Yassin Chakhchoukh

Laut Richard Bartle geht es bei vielen **Browser- und Social Games** nicht um die niedliche Kuh oder das zahme Tigerchen, sondern nur darum, wie viel Geld beide einbringen.

Moderne Rollenspiele wie **World of Warcraft** sehen sich anders als die MUDs einem »Raumproblem« gegenüber. Sie können nicht einfach riesige Städte oder Kontinente anbieten, der Spieler muss diese auch tatsächlich in angemessener Zeit durchqueren können.

Ein Tag beim ...
Sims Studio

Der riesige EA-Campus beherbergt auch das Gebäude, in dem täglich an den Sims-Spielen gearbeitet wird.

Auch aus der Luft ist das Sims-Hauptquartier sofort erkennbar.

Einige Sims-Mitarbeiter beim spontanen Erschrecken vor einer Dragon-Age-Statue.

Associate Producer MJ Chun bei den Aufnahmen für ein Making-Of.

Das EA Campus Fitness Center – hier wird kostenlos gestemmt.

Audio Director Robi Kauker haut in die Piano-Tasten.

An EAs »Wall of Games« darf das original »Die Sims« natürlich nicht fehlen.

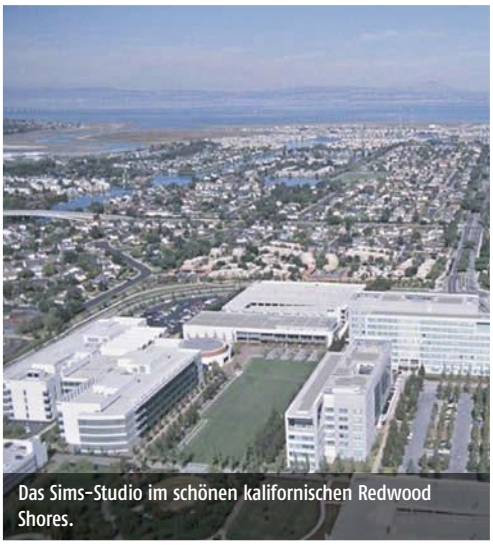
Lead Character Modeller Steve Rheinfrank perfektioniert die Gesichtsanimationen für »Die Sims Mittelalter«.

Graphic Design Artist Aurora Jimenez beim Feinschliff für neue Mittelalter-Charaktere.

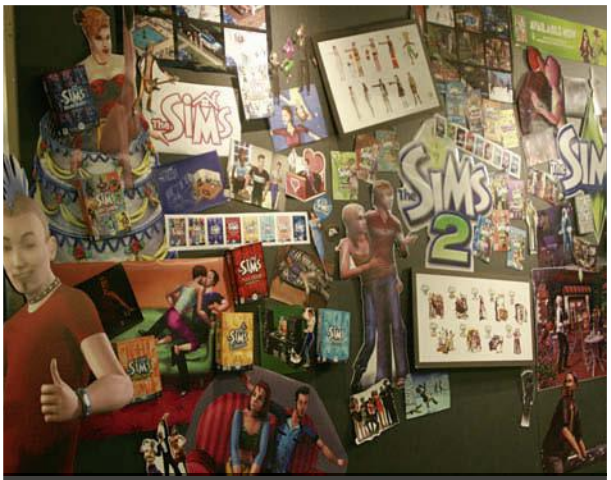
Dan Hiatt entspannt sich beim Laute spielen.

Im Februar feierten »Die Sims« ihr zehnjähriges Jubiläum – Grund genug, einmal den Entwicklern im kalifornischen Redwood Shores über die Schulter zu blicken. Die Erkenntnis: Sie dekorieren gern. Und sie feiern gern.

»Die Sims 3: Late Night«-Producer Grant Rodiek an seinem gut sortierten Arbeitsplatz.

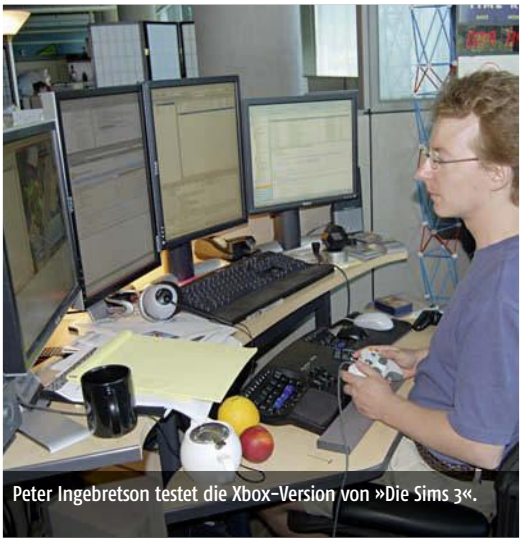


Das Sims-Studio im schönen kalifornischen Redwood Shores.



Im Februar feierten »Die Sims« ihr zehnjähriges Jubiläum, diese Wand zeigt die Errungenschaften der Erfolgsserie.

Kostbare Original-Artworks zum Sims-Geburtstag.



Peter Ingebretson testet die Xbox-Version von »Die Sims 3«.

Hier entsteht gerade das Design eines Raumes für »Die Sims Mittelalter«.

Alle zwei Wochen feiern wir »Thank God It's Friday«-Party.

Das gesamte Sims-Team beim Photoshoot auf dem »EA Redwood Shores«-Rasen.

Stilecht wird der Start von »Die Sims 3: World Adventures« auf einer Bowlingbahn gefeiert – natürlich inklusive Pizza.

In der nächsten Ausgabe
Making Games Vorschau

Making Games 02/2011

erscheint am 25. Februar 2011

Titelthema:

Das Brettspiel-Comeback

Carcassone rockt die iPhone-Charts, Ravensburger verstärkt sein Engagement im Digitalbereich, Die Siedler 7 setzt auf Brettspiel-Game-Design. Was macht den Erfolg der Brettspiel-Umsetzungen aus, was sind die Trends und Marktchancen? Und was können selbst Rollenspiele und Shooter von ihnen lernen? Wir liefern die Antworten.

Weitere Themen:

- **Interview**
  DLC: Paul Hossack von EA über den Produktionsaufwand von Fifa Ultimate Team.
- **Programmierung**
  Screenspace Antialiasing: So funktioniert die Kantenglättung von SpecOps.
- **Forschungsstudie**
  Die TU Ilmenau untersucht, wie sich der Gebrauchtspiele-Markt entwickeln wird.

IMPRESSUM

Chefredakteur
Heiko Klinge
(verantwortlich, Anschrift d. Red.)

Redaktion
Gunnar Lott

Redaktion
Markus Schwerdtel

Redaktion
Yassin Chakhchoukh

Redaktion
Christian Schmidt

Lektorat (Freelancer)
Marion Schneider

Layout (ltd.)
Sigrun Rüb

Layout
Konstantin Christanakis

Layout
Manfred Aumaier

Layout (Trainee)
Oliver Thomas

Layout (Freelancer)
Eva Zechmeister

Freie Mitarbeiter dieser Ausgabe (Redaktion):
René Bauer, Daniel Dumont, Florian Faller, Frank Fay, Prof. Ulrich Götz, Olliver Heins, Oliver Iking, Sina Konz, Andy Moore, Cornelius Müller, Reto Spoerri, Alexander Taube, Jochen Winkler, Richard Winterstetter

Wir danken den Interview-Partnern:
Paul Bakaus, Richard Bartle, Nick Burton, Rocco Di Leo, Thomas Friedmann, Nicola Godin, Tau Petersson

Titelgrafik: h2design.de

Assistenz:
Isa Stamp, Anita Thiel, Annie Weissenberger

Korrespondenten:
Stacy Cowley (New York), Zhang Ran (Shanghai), Martyn Williams (Tokio), Peter Sayer (Paris), Sumner Lemon (Taipeh), George Dragon (Budapest), Lukas Erben (Prag)

Kundenservice

Abonnement und Einzelbestellung:
Making Games Kundenservice
ZENIT Pressevertrieb GmbH
Postfach 81 05 80, 70522 Stuttgart
Tel.: 01805 / 727252-275
Fax: 01805 / 727252-377
Österreich: Tel.: 01 / 219 55 60
Schweiz: Tel.: 071 / 314 06-15
E-Mail: shop@makinggames.de
Web: www.makinggames.de/shop
Bezugspreise:
Einzelpreis: 6,90 Euro
Jahresabo: 28,98 Euro
(beinhaltet 6 Ausgaben)
Auslandspreise u. Mengenrabatt auf Anfrage
Zahlungsmöglichkeit:
Postbank Stuttgart, BLZ 600 100 70
Konto-Nr.: 311 704
So erreichen Sie die Redaktion:
IDG Entertainment Media GmbH
Redaktion Making Games
Lyonel-Feininger-Straße 26
80807 München
Tel.: 089 / 360 86-660, Fax: 089 / 360 86-652

Anzeigenkontakt

Sales Manager Making Games (Freelancer):
Nicole Müller, 089 / 120 353 23
nmueller@idg-consultant.de
Anzeigenleiter (verantwortlich):
Ralf Sattelberger, 089 / 360 86 (-730)
Senior Sales Manager:
Annika Gredelewski (-625)
agradelewski@gamestar.de
Junior Account Manager:
Susanne Fütterer (-670)
sfuetterer@gamestar.de
Sales Service Manager:
Daniela Kretschek (-691) Fax: (-672)
Digitale Anzeigenannahme Print:
Manfred Aumaier (-602)
anzeigendisposprint@gamestar.de
Digitale Anzeigenannahme Online:
IDG Sites E-Mail Sammeladresse:
banner@idg.de
Anzeigenpreise:
Es gilt die Preisliste gültig ab 01.01.2010

Vertrieb Handelsauflage

MZV GmbH & Co. KG
Ohmstraße 1, 85716 Unterschleißheim
Telefon: 089 / 319 06-0, Fax: -113
E-Mail: info@mzv.de
Web: www.mzv.de

Verlag

IDG Entertainment Media GmbH
Lyonel-Feininger-Straße 26
80807 München
Tel.: 089 / 360 86-0, Fax: 089 / 360 86-501
www.idgmedia.de
Geschäftsführer: York von Heimburg
Verlagsleitung: André Horn
Leiter CRM/Marketing: Matthias Weber (-154)
Veröffentlichung gemäß §8, Absatz 3 des Gesetzes über die Presse vom 08.10.1949: Alleiniger Gesellschafter der IDG Entertainment Media GmbH ist die IDG Communications Media AG, München, die 100% Tochter der International Data Group Inc., Boston, USA, ist.
Vorstand:
York von Heimburg, Keith Arnot, Bob Carrigan
Aufsichtsratsvorsitzender:
Patrick J. McGovern
Produktionsleitung: Jutta Eckebrecht
Druck und Beilagen:
Mayr Miesbach GmbH, Druckerei und Verlag
83714 Miesbach

Haftung & Co.

© Copyright IDG Entertainment Media GmbH
Haftung: Für die Richtigkeit von Veröffentlichungen können Redaktion und Verlag trotz Prüfung nicht haften. Die Veröffentlichungen in Making Games erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Auch werden Warennamen ohne Gewährleistung einer freien Anwendung benutzt.

qantm INSTITUTE

Ausbildungsinstitut für
Gamedesign & Development,
Animation und kreative Medien

NL • GER • AU • UK • SG • AT •CH

KURSSTART:
März 2011 &
September 2011

INFOTAG:
30. Januar 2011
14:00 - 18:00 Uhr

www.qantm.de • www.qantm.tv • www.sae.edu
Mail: berlin@qantm.de • muenchen@qantm.de
Berlin: +49 (0)30 43094470 München: +49 (0)89 89068771

DIPLOMA & BACHELOR*
OF INTERACTIVE ENTERTAINMENT

Middlesex University

* in Kooperation mit der Middlesex University of London

Köln und Fußball schließen sich nicht aus.

Jetzt bewerben!

Wir bieten interessante und abwechslungsreiche Tätigkeiten in spektakulären Büros im Kölner Rheinauhafen. Internationalität wird bei uns großgeschrieben, und seit mittlerweile fünf Jahren arbeiten wir erfolgreich mit deutschen, englischen und kanadischen Teams zusammen. Neben dem Fußballmanager liegt unser zweiter Entwicklungsschwerpunkt bei Browserspielen.

Schriftliche Bewerbungen oder Bewerbungen per Mail (mit Gehaltsvorstellungen) bitte an:

Bright Future GmbH
z.H. Gerald Köhler
Im Zollhafen 15-17
50678 Köln
jobs@brightfuture.de

Weitere Informationen über Bright Future und unsere Produkte finden Sie auf www.brightfuture.de.

bright future
Bright Future GmbH